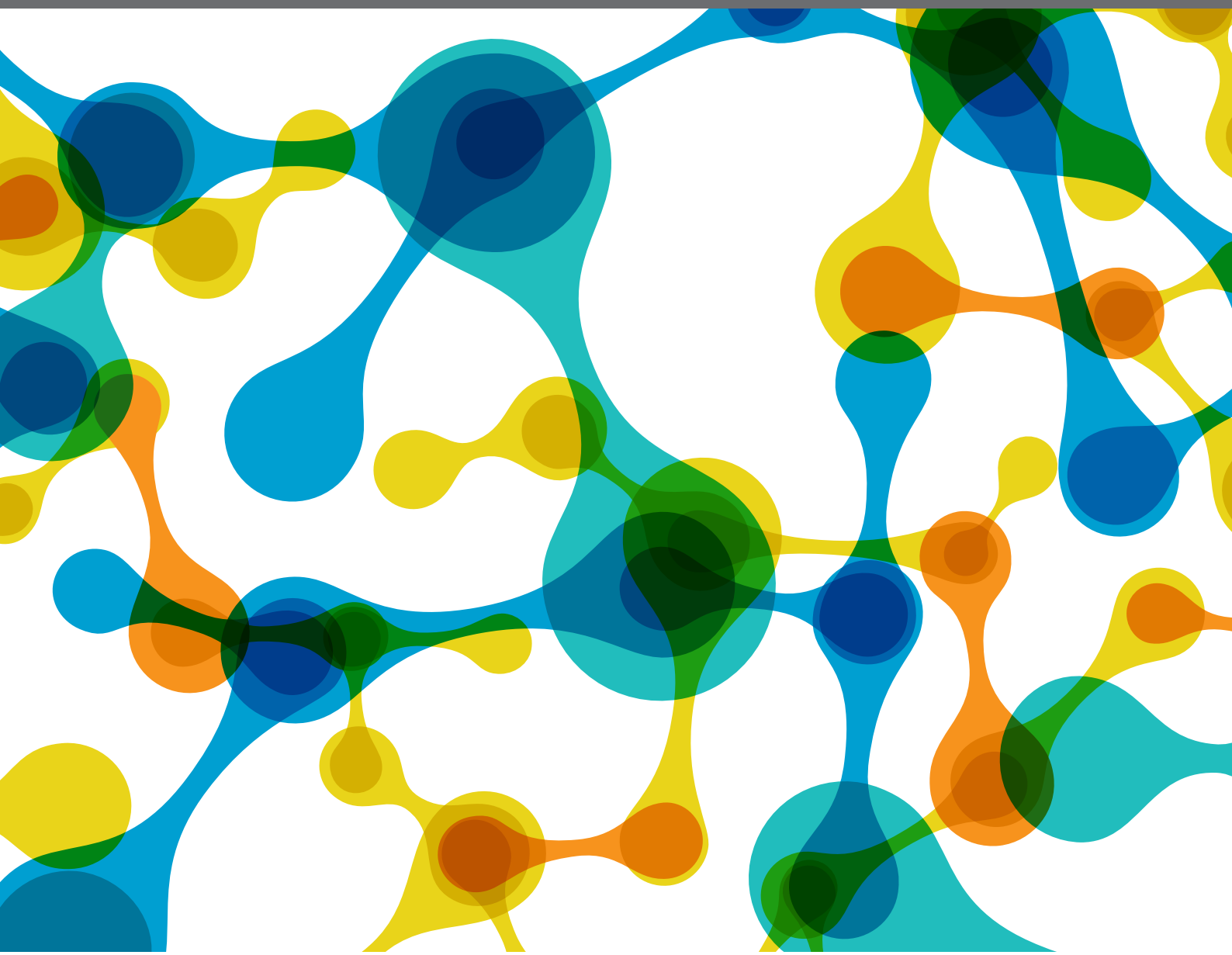
LOGICAL MODELING OF CELLULAR PROCESSES: FROM SOFTWARE DEVELOPMENT TO NETWORK DYNAMICS

EDITED BY: Matteo Barberis and Tomáš Helikar
PUBLISHED IN: Frontiers in Physiology



frontiers Research Topics

ISSN 1664-8714
ISBN 978-2-88945-983-4
DOI 10.3389/978-2-88945-983-4

About Frontiers

Frontiers is more than just an open-access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers Journal Series

The Frontiers Journal Series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the Frontiers Journal Series operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to Quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews.
Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the Frontiers Journals Series: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area! Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers Editorial Office: researchtopics@frontiersin.org

LOGICAL MODELING OF CELLULAR PROCESSES: FROM SOFTWARE DEVELOPMENT TO NETWORK DYNAMICS

Topic Editors:
Matteo Barberis, University of Surrey, United Kingdom; University of Amsterdam, Netherlands
Tomáš Helikar, University of Nebraska-Lincoln, United States

Mathematical models have become invaluable tools for understanding the intricate dynamic behavior of complex biochemical and biological systems. Among computational strategies, logical modeling has been recently gaining interest as an alternative approach to address network dynamics. Due to its advantages, including scalability and independence of kinetic parameters, the logical modeling framework is becoming increasingly popular to study the dynamics of highly interconnected systems, such as cell cycle progression, T cell differentiation and gene regulation. Novel tools and standards have been developed to increase the interoperability of logical models, which can now be employ to respond a variety of biological questions. This Research Topic brings together the most recent and cutting-edge approaches in the area of logical modeling including, among others, novel biological applications, software development and model analysis techniques.

Citation: Barberis, M., Helikar, T., eds. (2019). Logical Modeling of Cellular Processes: From Software Development to Network Dynamics. Lausanne: Frontiers Media. doi: 10.3389/978-2-88945-983-4

Table of Contents

05 ***A Network Model to Explore the Effect of the Micro-environment on Endothelial Cell Behavior During Angiogenesis***
Nathan Weinstein, Luis Mendoza, Isidoro Gitler and Jaime Klapp

23 ***Automatic Screening for Perturbations in Boolean Networks***
Julian D. Schwab and Hans A. Kestler

31 ***Target Control in Logical Models Using the Domain of Influence of Nodes***
Gang Yang, Jorge Gómez Tejeda Zañudo and Réka Albert

48 ***Using Regularization to Infer Cell Line Specificity in Logical Network Models of Signaling Pathways***
Sébastien De Landtsheer, Philippe Lucarelli and Thomas Sauter

61 ***DSGRN: Examining the Dynamics of Families of Logical Models***
Bree Cummins, Tomas Gedeon, Shaun Harker and Konstantin Mischaikow

69 ***Analysis Tools for Interconnected Boolean Networks With Biological Applications***
Madalena Chaves and Laurent Tournier

87 ***Identification of Boolean Network Models From Time Series Data Incorporating Prior Knowledge***
Thomas Leifeld, Zhihua Zhang and Ping Zhang

99 ***Logical Modeling and Analysis of Cellular Regulatory Networks With GINsim 3.0***
Aurélien Naldi, Céline Hernandez, Wassim Abou-Jaoudé, Pedro T. Monteiro, Claudine Chaouiya and Denis Thieffry

115 ***The CoLoMoTo Interactive Notebook: Accessible and Reproducible Computational Analyses for Qualitative Biological Networks***
Aurélien Naldi, Céline Hernandez, Nicolas Levy, Gautier Stoll, Pedro T. Monteiro, Claudine Chaouiya, Tomáš Helikar, Andrei Zinovyev, Laurence Calzone, Sarah Cohen-Boulakia, Denis Thieffry and Loïc Paulevé

128 ***Prediction of Mutations to Control Pathways Enabling Tumor Cell Invasion With the CoLoMoTo Interactive Notebook (Tutorial)***
Nicolas Levy, Aurélien Naldi, Céline Hernandez, Gautier Stoll, Denis Thieffry, Andrei Zinovyev, Laurence Calzone and Loïc Paulevé

140 ***Global Stabilization of Boolean Networks to Control the Heterogeneity of Cellular Responses***
Jung-Min Yang, Chun-Kyung Lee and Kwang-Hyun Cho

157 ***A Mechanistic Computational Model Reveals That Plasticity of CD4$^+$ T Cell Differentiation is a Function of Cytokine Composition and Dosage***
Bhanwar Lal Puniya, Robert G. Todd, Akram Mohammed, Deborah M. Brown, Matteo Barberis and Tomáš Helikar

175 ***Role of Cytokine Combinations on CD4+ T Cell Differentiation, Partial Polarization, and Plasticity: Continuous Network Modeling Approach***
Mariana E. Martinez-Sanchez, Leonor Huerta, Elena R. Alvarez-Buylla and Carlos Villarreal Luján

189 ***CANA: A Python Package for Quantifying Control and Canalization in Boolean Networks***
Rion B. Correia, Alexander J. Gates, Xuan Wang and Luis M. Rocha

197 ***Identification of Biologically Essential Nodes via Determinative Power in Logical Models of Cellular Processes***
Trevor Pentzien, Bhanwar L. Puniya, Tomáš Helikar and Mihaela T. Matache

217 ***Estimating Attractor Reachability in Asynchronous Logical Models***
Nuno D. Mendes, Rui Henriques, Elisabeth Remy, Jorge Carneiro, Pedro T. Monteiro and Claudine Chaouiya

232 ***Dynamics of the Gene Regulatory Network of HIV-1 and the Role of Viral Non-coding RNAs on Latency Reversion***
Antonio Bensussen, Christian Torres-Sosa, Ramón A. Gonzalez and José Díaz

250 ***Evaluating Uncertainty in Signaling Networks Using Logical Modeling***
Kirsten Thobe, Christina Kuznia, Christine Sers and Heike Siebert

265 ***BioLQM: A Java Toolkit for the Manipulation and Conversion of Logical Qualitative Models of Biological Networks***
Aurélien Naldi

275 ***Gene Regulatory Network Modeling of Macrophage Differentiation Corroborates the Continuum Hypothesis of Polarization States***
Alessandro Palma, Abdul Salam Jarrah, Paolo Tieri, Gianni Cesareni and Filippo Castiglione

294 ***Modeling the Role of the Microbiome in Evolution***
Saúl Huitzil, Santiago Sandoval-Motta, Alejandro Frank and Maximino Aldana

308 ***Robustness of Nutrient Signaling is Maintained by Interconnectivity Between Signal Transduction Pathways***
Niek Welkenhuysen, Barbara Schnitzer, Linnea Österberg and Marija Cvijovic

321 ***Personalization of Logical Models With Multi-Omics Data Allows Clinical Stratification of Patients***
Jonas Béal, Arnau Montagud, Pauline Traynard, Emmanuel Barillot and Laurence Calzone

frontiers in Physiology

ORIGINAL RESEARCH
published: 27 November 2017
doi: 10.3389/fphys.2017.00960

A Network Model to Explore the Effect of the Micro-environment on Endothelial Cell Behavior during Angiogenesis

Nathan Weinstein[1]*, *Luis Mendoza*[2], *Isidoro Gitler*[1] and *Jaime Klapp*[1, 3]*

[1] ABACUS-Laboratorio de Matemáticas Aplicadas y Cómputo de Alto Rendimiento, Departamento de Matemáticas, Centro de Investigación y de Estudios Avanzados CINVESTAV-IPN, Mexico City, Mexico, [2] CompBioLab, Departamento de Biología Molecular y Biotecnología, Instituto de Investigaciones Biomédicas, Universidad Nacional Autónoma de México, Mexico City, Mexico, [3] Departamento de Física, Instituto Nacional de Investigaciones Nucleares, Mexico City, Mexico

OPEN ACCESS

Edited by:
Matteo Barberis,
University of Amsterdam, Netherlands

Reviewed by:
Laurence Calzone,
Institut Curie, France
Aleksander S. Popel,
Johns Hopkins University,
United States

***Correspondence:**
Nathan Weinstein
nathan.weinstein4@gmail.com
Jaime Klapp
jaime.klapp@inin.gob.mx

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: 07 September 2017
Accepted: 10 November 2017
Published: 27 November 2017

Citation:
Weinstein N, Mendoza L, Gitler I and Klapp J (2017) A Network Model to Explore the Effect of the Micro-environment on Endothelial Cell Behavior during Angiogenesis. Front. Physiol. 8:960. doi: 10.3389/fphys.2017.00960

Angiogenesis is an important adaptation mechanism of the blood vessels to the changing requirements of the body during development, aging, and wound healing. Angiogenesis allows existing blood vessels to form new connections or to reabsorb existing ones. Blood vessels are composed of a layer of endothelial cells (ECs) covered by one or more layers of mural cells (smooth muscle cells or pericytes). We constructed a computational Boolean model of the molecular regulatory network involved in the control of angiogenesis. Our model includes the ANG/TIE, HIF, AMPK/mTOR, VEGF, IGF, FGF, PLCγ/Calcium, PI3K/AKT, NO, NOTCH, and WNT signaling pathways, as well as the mechanosensory components of the cytoskeleton. The dynamical behavior of our model recovers the patterns of molecular activation observed in Phalanx, Tip, and Stalk ECs. Furthermore, our model is able to describe the modulation of EC behavior due to extracellular micro-environments, as well as the effect due to loss- and gain-of-function mutations. These properties make our model a suitable platform for the understanding of the molecular mechanisms underlying some pathologies. For example, it is possible to follow the changes in the activation patterns caused by mutations that promote Tip EC behavior and inhibit Phalanx EC behavior, that lead to the conditions associated with retinal vascular disorders and tumor vascularization. Moreover, the model describes how mutations that promote Phalanx EC behavior are associated with the development of arteriovenous and venous malformations. These results suggest that the network model that we propose has the potential to be used in the study of how the modulation of the EC extracellular micro-environment may improve the outcome of vascular disease treatments.

Keywords: sprouting angiogenesis, network model, mechanical stress, cell differentiation, cell polarization, lateral inhibition

1. INTRODUCTION

The circulatory system allows for the existence of large multicellular organisms, ensuring adequate oxygen and nutrient supply. Blood vessels are composed of three main layers. The outermost layer—the tunica adventitia—contains elastic fibers, collagen, and connective tissue. The middle layer—the tunica media—is comprised of smooth muscle cells, collagen, and elastin, and the innermost layer—the tunica intima—, which is exposed to the vessel lumen, is a single-cell layer of endothelium. The circulatory system is not a static structure, it adapts to the changing requirements of the body by means of vasculogenesis, arteriogenesis, and angiogenesis (Betz et al., 2016).

Vasculogenesis is a process that allows for the *de novo* formation of blood vessels. The formation of the first blood vessels in the embryo involves the differentiation of cells from the mesodermal blood islands into angioblasts, also called endothelial precursor cells (EPCs). During later development, angioblasts may differentiate from hematopoietic stem cells, multipotent bone marrow progenitor cells, myeloid cells (specifically monocytes and macrophages), side population cells, and pluripotent stem cells (Kässmeyer et al., 2009). After the differentiation of EPCs, the cells must migrate and aggregate to form a primitive vascular blood plexus. Then, for the vascular network assembly, three mechanisms have been proposed: (a) Extracellular matrix contact guidance, where the ECs are guided by collagen fibers present in the extracellular matrix and each cell may change the tension and orientation of the collagen fiber network to guide other cells, (b) Autocrine chemotaxis, where the ECs follow a morphogen (such as VEGFA) gradient and then secrete the morphogen altering the gradient to guide other cells, and (c) Cell-to-cell contact, where sprout expansion is guided by contact with multicellular elongated structures or projections of other cells (Czirok, 2013).

Arteriogenesis increases the diameter of existing blood vessels and remodels large blood vessels creating natural bypasses when necessary. Whenever, blood flow is redirected into preexisting arterioles, it creates mechanical forces. Elevated shear stress and circumferential wall stress during a long time period are strong inducers of arteriogenesis (Heil et al., 2006). The endothelium of the arteriolar connections is activated by the mechanical forces, causing monocytes to promote arteriogenesis by secreting growth factors and cytokines that increase the mitosis rate of endothelial and smooth muscle cells (Deindl and Schaper, 2005). Perivascular mast cells mediate shear stress-induced arteriogenesis by coordinating the action of T cells, neutrophils, monocytes, macrophages, and other innate immune cells by means of the secretion of cytokines and MMPs. The activation of perivascular mast cells is achieved by the increase of Nox2-derived reactive oxygen radicals, caused by neutrophil extravasation (Chillo et al., 2016).

Angiogenesis extends, maintains, and remodels existing networks of thin blood vessels, mostly capillaries. There exist two main mechanisms for angiogenesis, namely, sprouting angiogenesis (SA), and splitting angiogenesis, also known as intussusceptive angiogenesis (IA) (Gianni-Barrera et al., 2011). Alterations in blood flow and local changes in the concentration of angiogenic factors such as VEGF may trigger angiogenesis. Laminar shear stress inhibits tubule formation and migration of endothelial cells and favors intussusceptive angiogenesis, while turbulent shear stress causes an increase in cell migration and proliferation, and favors sprouting angiogenesis (Makanya et al., 2009). In skeletal muscle, $VEGFA_{164}$ over-expression induces vascular growth by intussusception rather than sprouting (Gianni-Barrera et al., 2013).

IA occurs during physiological adaptation i.e., exercised muscles, embryonic development, and pathological situations such as tumor growth. During IA, endothelial cells extend processes into the vascular lumen from opposing walls. Once these processes contact each other, the endothelial cell junctions at the contact site are reorganized. Then, the bilayer is perforated by invading interstitial tissue, pericytes, and myofibroblasts, forming a transluminal pillar. Subsequently, pericytes, fibroblasts, and other supporting cells deposit additional collagen and other stabilizing fibers into the extracellular matrix of the pillar (Makanya et al., 2009), that may increase in girth, until it splits the blood vessel into two independent vascular segments (Patan et al., 1996, 1997). Additionally, several transluminal pillars may fuse to split a vessel or improve local hemodynamic behavior (Kurz et al., 2003). IA has three main advantages over SA: first, IA is achieved with minimal tissue degradation and reduced vascular permeability caused by mural cell detachment, second, a relatively short period of time is sufficient to achieve it, and third, only a relatively low rate of endothelial proliferation is needed (Kurz et al., 2003; Makanya et al., 2009; Gianni-Barrera et al., 2011). IA is necessary for the formation of organ-specific angioarchitecture (intussusceptive microvascular growth), the formation of vascular trees (intussusceptive arborization), angioadaptation

Abbreviations: AMP, Adenosine Monophosphate; AMPK, AMP-activated Protein Kinase; ANG, Angiopoietin; APLN, Apelin; ASF, Alternative Splicing Factor; ATP, Adenosine Triphosphate; BMP, Bone Morphogenetic Protein; BTrCP, Beta-Transducin Repeat Containing E3 Ubiquitin Protein Ligase; CXCR4, C-X-C motif chemokine Receptor 4; DAAM, Dishevelled Associated Activator Of Morphogenesis; DLL, Delta-Like canonical notch Ligand; DSH, Dishevelled; EC, Endothelial Cell; ECM, Extracellular Matrix; eNOS, Endothelial Nitric Oxide Synthase; EPC, Endothelial Progenitor Cells; EPH, Ephedrin; ERK, Extracellular signal-Regulated Kinase; FAK, Focal Adhesion Kinase; FOXO1, Forkhead Box O1; FGF, Fibroblast Growth Factor; FZD, Frizzled; GSK3β, Glycogen Synthase Kinase 3 Beta; HSC, Hematopoietic Stem Cell; HEY, Hes related family BHLH transcription factor with YRPW motif; HIF, Hypoxia-Inducible Factor; IA, Intussusceptive Angiogenesis; IGF, Insulin-like Growth Factor; JAG, Jagged; KLF, Kruppel Like Factor; LRP, LDL Receptor Related Protein; LEF, Lymphoid Enhancer binding Factor; MAPK, Mitogen-Activated Protein Kinase; MEK, Mitogen-Activated Protein Kinase Kinase (MAPKK); MMP, Matrix Metalloproteinase; mTOR, mechanistic Target Of Rapamycin; NFAT, Nuclear Factor of Activated T-cells; NICD, NOTCH Intracellular Domain; NO, Nitric Oxide; Nox2, NADPH oxidase 2; NRARP, NOTCH Regulated Ankyrin Repeat Protein; NRP1, Neuropilin 1; PA, Plasminogen Activator; PDGF, Platelet-Derived Growth Factor; PECAM, Platelet and Endothelial Cell Adhesion Molecule; PI3K, Phosphatidylinositol-4,5-bisphosphate 3-Kinase; PIP3, Phosphatidylinositol (3,4,5)-trisphosphate; PKC, Protein Kinase C; PTEN, Phosphatase and Tensin homolog; RHO, Rhodopsin; SA, Splitting Angiogenesis; SC, Stalk Cell; SF2, pre-mRNA-Splicing Factor 2; SIRT, Sirtuin, S1P, sphingosine-1-Phosphate; S1PR, sphingosine-1-Phosphate Receptor; TC, Tip Cell; TGF, Transforming Growth Factor; TIE, Tyrosine kinase with domains similar to Immunoglobulin and Epidermal growth factor; TSC, Tuberous Sclerosis; uPAR, Urokinase Receptor; VEGF, Vascular Endothelial Growth Factor; VEGFR, Vascular Endothelial Growth Factor-Receptor.

and vascular pruning (intussusceptive branching remodeling) (Makanya et al., 2009).

SA is a developmental process that results in a new connection between two existing thin blood vessels (**Figure 1**) and involves eight related cellular processes: (1) *Secretion of angiogenic factors*. Shear stress, or an insufficient local supply of oxygen or nutrients, may cause the cells within a tissue to secrete angiogenic factors (Forsythe et al., 1996; Song and Munn, 2011; Kumar et al., 2014). Relevant angiogenic factors include growth factors, chemokines, angiopoietins, endostatin, interferons, and NO among other molecules (Logsdon et al., 2014). (2) *Vessel destabilization*. Before a new sprout may form, pericytes, myofibroblasts, and other supporting cells must be cleared from the area of the blood vessel where the sprout will form. Also, the ECM surrounding the area must be remodeled. Blood vessel destabilization is mediated by VEGFA, ANG2, NO, and the absence of blood flow (Scharpfenecker et al., 2005; Qin et al., 2013; Korn and Augustin, 2015). (3) *Tip and stalk cell differentiation*. When certain ECs are exposed to a VEGF gradient some respond to VEGFA and shear stress to become tip cells (TCs), growing filopodia toward the VEGFA gradient. TCs induce neighbor cells to become stalk cells (SCs) by Notch-mediated lateral signaling (Blanco and Gerhardt, 2013). TCs become less sensitive to Notch signaling and SCs become less sensitive to VEGF signaling (Weinstein et al., 2015; Glass et al., 2016). (4) *Sprout elongation*. The sprout is initially formed by the TC and one or two adjacent SCs. The subsequent proliferation of both the TC and SCs together with SC elongation and rearrangement support stalk elongation toward the VEGFA source resulting in stalk growth (Betz et al., 2016). (5) *Lumen formation and expansion*. Lumen formation may occur through cord hollowing, cell hollowing, trans-cellular lumen formation, and lumen ensheathment. Hemodynamic forces shape the apical membrane of SCs to form and expand new lumenized vascular tubes (Betz et al., 2016). (6) *Anastomosis*. Vascular anastomosis is the process that allows angiogenic sprouts and blood vessels to connect. Anastomosis can occur between two sprouts, or between a sprout and a functional blood vessel. The first step in an anastomosis is the formation of a stable contact between two ECs forming a new adherens junction with two layers of apical membrane and a small luminal volume in between. The mechanism that allows the formation of a new multicellular, perfused tubes depends on the presence or absence of blood pressure (Betz et al., 2016). (7) *Vessel stabilization*. Once a lumenized new blood vessel has formed, ECs release platelet-derived growth factor B (PDGFB). PDGFB attracts pericytes, which incorporate into the vessel wall. S1P, S1PR1, ANG1, TIE2, Ephrin-B2, EPH, and TGFβ regulate blood vessel stabilization and maturation and are regulated by shear stress (Scharpfenecker et al., 2005; Qin et al., 2013; Korn and Augustin, 2015). And (8) *Pruning*. Vessel pruning is basically the process of sprout

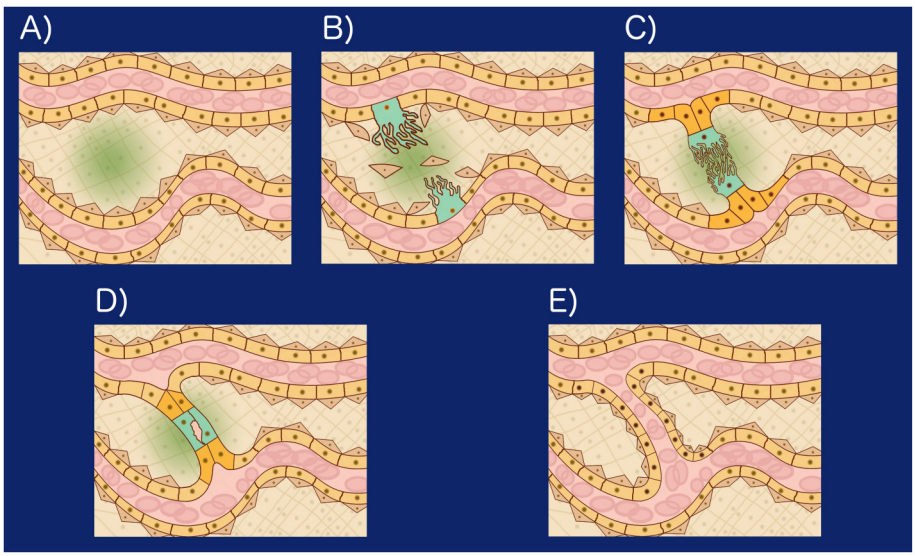


FIGURE 1 | (A) Hypoxia induced angiogenesis: When tissue cells are exposed to a microenvironment with an insufficient concentration of Oxygen, they secrete VEGFA in a process mediated by the Hypoxia-inducible factor 1 (HIF-1). Forming a VEGFA gradient (green), **(B)** Certain epithelial cells (peach) respond to VEGFA and shear stress to become tip cells (TCs): VEGFA, ANG2, shear stress, and NO lead to endothelial cell matrix degradation, loss of pericytes (brown triangular cells). Certain EC become TCs (turquoise) and grow filopodia toward the VEGFA gradient. TCs inhibit neighboring cells from becoming TCs by Notch mediated lateral signaling and Wnt, **(C)** Stalk growth and anastomosis: The cells neighboring the TCs are induced by Notch to become Stalk cells (SCs). SCs (orange) secrete VEGFR1, reducing the concentration of VEGFA in their microenvironment, undergo Wnt mediated proliferation and elongate toward the VEGFA source resulting in stalk growth. Once a TC reaches another TC or vessel wall, it undergoes VE-cadherin and Macrophage mediated binding, initiating anastomosis, **(D)** Lumen formation: Lumen formation may occur through cord hollowing (Intracellular vacuoles fuse intracellularly to hollow out stalk cells and generate an interconnected luminal space), cell hollowing, transcellular lumen formation, and lumen ensheatment. Hemodynamic forces shape the apical membrane of SCs to form and expand new lumenized vascular tubes, **(E)** Vessel stabilization: Once a lumenized new blood vessel has formed, ECs release platelet-derived growth factor B (PDGFB). PDGFB attracts pericytes which incorporate into the vessel wall. S1P, S1PR1, ANG1, TIE2, Ephirin-B2, EPH, and TGF regulate blood vessel stabilization and maturation and are regulated by shear stress.

formation in reverse. The absence of blood flow, or a higher anti-angiogenic (ANG2) to angiogenic (VEGFA) factor ratio, induces small blood vessel pruning by reabsorption of ECs into the remaining vasculature. Regression of larger blood vessels involves apoptosis (Korn and Augustin, 2015; Betz et al., 2016).

Due to the enormous biological and medical importance of angiogenesis, many computational and mathematical models have been proposed to explore the molecular mechanism involved in angiogenesis control (Peirce, 2008; Qutub et al., 2009; Scianna et al., 2013; Logsdon et al., 2014; Heck et al., 2015; Qutub and Popel, 2015). Some of the relevant previous models are the following: (a) A computational model exploring the relationship between hemodynamics and angiogenesis in 2D (Gödde and Kurz, 2001). (b) A computational model of oxygen transport in skeletal muscle for sprouting and splitting modes of angiogenesis (Ji et al., 2006). (c) A model that describes and explores the progression of angiogenesis during the healing process (Vermolen and Javierre, 2012). (d) A multicellular model of the early stages of angiogenesis using finite element integration that includes chemotaxis and the interaction between tip cells and stalk cells (Bookholt et al., 2016). (e) Two Boolean models that explore the relationship between the Wnt and VEGF signaling pathways, mechanoreceptors, apoptosis, cell proliferation, and lumen formation during angiogenesis (Bauer et al., 2010; Bazmara et al., 2016). And (f) a multilevel model based on the previously mentioned Boolean models (Bazmara et al., 2015). Notably, the authors of Bauer et al. (2010) and Bazmara et al. (2015) included apoptosis in their model. We did not include apoptosis in our model because thin blood vessel pruning usually occurs by the reabsorption of ECs into the remaining vasculature and seldom involves apoptosis.

Previous models of angiogenesis focused on the role played by a few of the canonical signaling pathways. However, recent discoveries have emphasized the role of TGF signaling and its interaction with the WNT, NOTCH, VEGF/NRP1, HIF, AKT, ERK, mTOR, and TIE signaling pathways, as well as the role of HIFs, Ca^{2+}, NO/eNOS, and cytoskeletal mechanoreceptors during angiogenesis. As a result, none of the previous models explore the interaction among all the aforementioned canonical pathways. It was not possible to know, then, if the biological system was sufficiently well characterized from the point of view of the molecular regulation. Hereby we present a model that integrates the largest set of canonical signaling pathways, thus allowing for a comprehensive characterization of the effect of the extracellular micro-environment on EC behavior during differentiation of ECs angiogenesis. The model presents a qualitative agreement with a large set of experimental published results, showing that the regulatory network is a faithful reconstruction of the central molecular mechanism controlling the cell behavior of endothelial cells during angiogenesis. This characteristic permits the use of the model not only to describe the wild-type development and adaptation process but also to propose targets for intervention in certain diseases. Specifically, our model suggests that favoring a micro-environment that induces Phalanx EC behavior may suffice to improve the treatment of vascular retinal disorders and vascular malformations. Thus, our model can be considered as a platform to study several molecular scenarios affecting angiogenesis.

2. MATERIALS AND METHODS

2.1. Molecular Basis of the Regulatory Network

To assemble our model of the molecular network involved in angiogenesis control, we first explored how each one of the main stages angiogenesis is regulated and then explored how the molecules involved in the control of each stage interact with those that regulate the other stages. We started by exploring how the ANG/TIE signaling pathway acts as a gatekeeper of EC quiescence. Next, we inquired into the mechanisms that allow lack of sufficient oxygen or nutrients to destabilize blood vessels and trigger the angiogenic process. Then, we probed the mechanism that allows certain EC to be more sensitive to angiogenic signals by regulating VEGFR activity. Later, we analyzed how VEGF signaling activates the signaling pathways ERK1/2, PI3K-AKT, SRC, and p38 MAPK, and additionally phosphorylates STATs. After that, we inquired into the mechanisms that allow mechanoreceptors to respond to shear stress and radial stress to regulate VEGF signaling. Our ensuing action was to scrutinize the mechanism that allows the VEGF, NOTCH, WNT, and TGF signaling pathways to interact and regulate tip and stalk EC behavior. Last, we explored the mechanism that allows NOTCH and WNT to regulate EC proliferation. All those molecular mechanisms, their interactions and some of the most relevant references that describe the experimental evidence are discussed in detail in the first section of the Supplementary Material.

2.2. The Regulatory Network as a Discrete Dynamical System

Boolean networks are discrete dynamical systems, whose simplicity allows the attainment of biologically meaningful results, after a systematic exploration of its dynamical behavior (Dubrova and Teslenko, 2011; Azpeitia et al., 2017). In our model, most variables represent genes or proteins, some represent small molecules, and one represents a mechanical force. Each variable has an activation state, that may be active, represented by a 1, or inactive, represented by a 0. Furthermore, we use a synchronous update approach where the states of all the variables are updated simultaneously. We decided to use a synchronous update scheme in our boolean model because the computational analysis of the asynchronous update is extremely time-consuming, and it is mostly required to explore race conditions and cyclic patterns of molecular activation (Garg et al., 2008; Saadatpour et al., 2010). However, neither race conditions nor cyclic behaviors are explored with our current model.

We use definitions and notation for Boolean networks based on Azpeitia et al. (2017). Let $\mathbb{B} = \{0, 1\}$ and $N^{+}_{\leq n} = \{1, 2, \ldots, n\}$, a set of labels. A *synchronous Boolean network with n components* is a function $f : \mathbb{B}^n \rightarrow \mathbb{B}^n$, where the *i-th component of* f is a function $f_i : \mathbb{B}^n \rightarrow \mathbb{B}$ such that $f_i(x) = f(x)_i$. A *state* of the network is an n-tuple $x = (x_1, x_2, \ldots, x_n)$ such that $x \in \mathbb{B}^n$. To

relate a synchronous Boolean network with a molecular network, we interpret that each component of a state x represents the activation state of a variable denoting a molecule included in the network. The dependency of the state on the discrete time parameter t is denoted as $x(t)$ and obeys the update rule given by f. That is for all $t \in \mathbb{Z}$

$$x(t+1) = f(x(t)) = (f_1(x(t)), f_2(x(t)), \ldots, f_n(x(t))),$$

where

$$x_i(t+1) = f_i(x(t)).$$

Our Boolean network model is deterministic, and any given initial state of the network reaches an attractor. A *fixed point attractor* is a state $s \in \mathbb{B}^n$ such that $f(s) = s$. We define f^{ol} as the l-th iterate of the function f such that $f^{ol} = f(f^{o(l-1)})$. An *attractor* is a set of states $A \subseteq \mathbb{B}^n$, such that $f^{ol}(x) = x$ for any state $x \in A$, in other terms, there exist a positive natural number $l \in \mathbb{N}^+ = \{1, 2, \ldots\}$ such that $f(x(t+l)) = f(x(t))$ for all $x(t) \in A$. Furthermore, l is the size of the attractor and for any $j \in \mathbb{N}^+_{<l}, f(x(t+j)) \in A$. Fixed point attractors represent stationary patterns of molecular activation, while larger attractors represent cyclic patterns of molecular activation. Additionally, we assume that each attractor represents an EC behavior.

For simplicity, we refer to the variable x_i by its position i in the n-tuple x. For a state $x \in \mathbb{B}^n$ and one of its components, say the one with label i, we denote by $x \sim i$ the n-tuple resulting from replacing the value of the variable x_i by its complement. Given two variables i and j and the update function of variable i, namely f_i, *j activates i* if there exists a pair of network states x, y that differ only in the state of activation of variable j, that is $y = x \sim j, x_j = 0$ and $y_j = 1$, such that $f_i(y) - f_i(x) > 0$. Conversely, *j inhibits i* if there exists a pair of network states x, y that differ only in the state of activation of variable j, that is $y = x \sim j, x_j = 0$ and $y_j = 1$, such that $f_i(y) - f_i(x) < 0$. An *interaction* denoted as the pair (i, j), $i, j \in \mathbb{N}_{\leq n}$ is *functional* if variable j activates or inhibits variable i. Note that according to this definition, it is possible for variable j to both activate and inhibit variable i depending on the functional context. For instance, let $C(t+1) = (A(t) \wedge \neg B(t)) \vee (\neg A(t) \wedge B(t))$. A activates C because if we focus on the cases where B is not active; if A is active, then C is active. A also inhibits C because if we focus on the cases where B is active; C is active only when A is not active.

2.3. Model Assembly

Using the information described in the subsection *Molecular basis of the network*, we assembled a model of the molecular network involved in angiogenesis control. Then we encoded the model using the standardized text file format required by BoolNet (Müssel et al., 2010), an R package for the analysis of Boolean networks. The models in BoolNet format, and the R scripts we used to simulate and analyze the dynamic behavior of the model are available at https://github.com/NathanWeinstein/Angiogenesis-Model/. During the development of our model, we ensured the existence of stable or cyclic patterns of molecular activation that correspond to Phalanx (AKT+, JAGa−, NRP1−), Stalk(JAGa+, NRP1−), and Tip (NRP+, DLL4a+, AKT−) EC behavior and their reachability under certain micro-environmental conditions (**Figure 4A**); specifically:

1. (*VEGFC_Dp*−, *VEGFAxxxP*−, *ANG1*+, *Oxygen*+, *ShearStress*+, *JAGp*−, *DLL4p*−, *WNT5a*−, *WNT7a*−, *FGF*−, *IGF*−, *BMP9*−, *BMP10*−, *TGFB1*−, *VEGFC_D*−, and *AMPATP*−) induces Phalanx EC behavior.
2. (*VEGFC_Dp*+, *VEGFAxxxP*−, *ANG1*+, *Oxygen*+, *ShearStress*+, *JAGp*−, *DLL4p*−, *WNT5a*−, *WNT7a*−, *FGF*−, *IGF*−, *BMP9*−, *BMP10*−, *TGFB1*−, *VEGFC_D*−, and *AMPATP*−) induces Tip EC behavior.
3. (*VEGFC_Dp*−, *VEGFAxxxP*+, *ANG1*+, *Oxygen*+, *ShearStress*+, *JAGp*−, *DLL4p*−, *WNT5a*−, *WNT7a*−, *FGF*−, *IGF*−, *BMP9*−, *BMP10*−, *TGFB1*−, *VEGFC_D*−, and *AMPATP*−) induces Tip EC behavior.
4. (*VEGFC_Dp*−, *VEGFAxxxP*−, *ANG1*+, *Oxygen*+, *ShearStress*+, *JAGp*−, *DLL4p*+, *WNT5a*+, *WNT7a*−, *FGF*−, *IGF*−, *BMP9*−, *BMP10*−, *TGFB1*+, *VEGFC_D*−, and *AMPATP*−) induces Stalk EC behavior.

Importantly, the expected patterns of molecular activation and EC behavior transitions are based on the literature (del Toro et al., 2010; Blancas et al., 2012; Glaser et al., 2016).

2.3.1. Simulation of an EC Behavior Transition

To simulate the transitions in EC behavior, we started with one of the states of an attractor that represents the initial EC behavior. Then, we modified the variables that represent the extracellular micro-environment (*VEGFC_Dp, VEGFAxxxP, ANG1, Oxygen, ShearStress, JAGp, DLL4p, WNT5a, WNT7a, FGF, IGF, BMP9, BMP10, TGFB1, VEGFC_D*, and *AMPATP*) without changing the other variables related to the internal state of the cell, to simulate a change of micro-environment that should lead to another EC behavior. We then calculated all state transitions until reaching another attractor that represents a new EC behavior.

2.3.2. Boolean Network Simplification

The size of the state space of a boolean molecular network represented as a directed graph with n nodes —one node for each variable—, grows exponentially as 2^n. Simulating and analyzing the dynamic behavior of networks containing more than a hundred nodes requires considerable computational resources. Recently, certain algorithms that use boolean satisfiability (SAT) methods capable of finding the attractors of networks with hundreds of nodes have been developed and implemented (Dubrova and Teslenko, 2011). Nonetheless, analyzing the effects of mutations, summarizing trap spaces, and analyzing the robustness of large networks is still a challenging task. However, it has been proved that it is possible to remove inputs and nodes with both an indegree and an outdegree equal to one without affecting the attractors (Saadatpour et al., 2013). Accordingly, we simplified the model by removing input nodes (nodes with an indegree equal to zero) that are either active, or inactive in all ECs, and are not part of the parameters that specify an extracellular micro-environment. Additionally, we removed output nodes (nodes with outdegree equal to zero). Further, we used edge contraction to merge intermediary nodes (nodes that have either an indegree or outdegree equal to one) that are not transcription

factors. The edge contraction operation involves the removal of an edge e (from u to v) and the merger of its two incident vertices, u and v, into a new vertex w. We assigned to w the name of u if v was only regulated by u, in this case we substituted $v(t)$ for $u(t)$ if e was positive or $\neg u(t)$ if e was negative in the components of f that correspond to the variables originally regulated by v. When u only regulated v, we assigned to w the name of v and in f_v we substituted $u(t)$ with f_u, that is, $f_v(\ldots, u(t), \ldots)$ becomes $f'_v(\ldots, f_u, \ldots)$. These operations allowed us to simplify our model without eliminating feedback circuits. This is relevant because to a large extent, feedback circuits determine the dynamic behavior of a boolean network (Azpeitia et al., 2017). The authors of Veliz-Cuba (2011) and Naldi et al. (2011) studied when the attractors are preserved after similar simplification processes. Additionally, we verified that the EC behaviors and transitions based on the literature were preserved after the simplification process. Further, we also verified that in both the detailed and the simplified model, all single gain and loss of function mutations have a similar effect on the EC behaviors and transitions based on the literature (Supplementary Figures 15, 16). Note that for this verification we only simulated the effect of 4 micro-environments. We only simulated the full effect of the mutations on our simplified model as part of the model validation process.

2.4. Analysis of the Dynamic Behavior of Our Model

First, we obtained all the attractors using the exhaustive SAT-based search available as part of BoolNet that uses an adaption of the algorithm proposed by Dubrova and Teslenko (2011). The exhaustive SAT-based search formulates the attractor search as a boolean satisfiability (SAT) problem that is solved using the PicoSAT solver (Biere, 2008). Then, we classified the attractors based on extracellular micro–environment. After that, for each micro-environment, we inferred the EC behavior represented by each attractor. If all EC behaviors associated to one micro-environment where of the same kind, we added that micro-environment to the set of micro-environments that induce that EC behavior. If not all EC behaviors associated with one micro-environment where of the same kind, we added the micro-environment to the set of micro-environments that induce atypical EC behavior. Finally, we summarized the four sets of micro-environments by grouping them into disjoint subsets using their shared characteristics. To validate our model, we simulated all single gain and loss of function mutations. We then compared the simulated effect of each mutation with its experimentally observed effect as reported in the literature (when available). Biological organisms need to be resilient to mutations and fluctuations in the concentration or level of molecular activation, we refer to this property as robustness. For clarity, it is necessary to indicate which trait is robust, to which perturbation and a method to quantify the resilience to define a robust feature (Félix and Barkoulas, 2015). We measured three robust features: (1) The robustness of Phalanx, Stalk, and Tip EC behaviors to single gain and loss-of-function mutations. This was measured as the percentage of mutations that prevent the existence of any stable or cyclic patterns of molecular activation that correspond to said EC behavior. (2) The robustness of attractor determination to molecular activation noise. First, we generated a set of 1,000,000 aleatory initial states. For each initial state, we created a perturbed copy with a Hamming distance of one by reversing the activatory state of one random variable. We quantified attractor determination robustness to molecular activation noise, as the fraction of the initial states that reached the same attractor as their perturbed copies. (3) The robustness of the trajectories that lead to Phalanx, Stalk, and Tip EC behaviors to molecular activation noise. First, we generated a set of 1,000,000 aleatory initial states. For each initial state, we created a perturbed copy with a Hamming distance of one by reversing the activatory state of one random variable. We quantified the robustness of the EC behaviors to molecular activation noise, as the fraction of the initial states that reached an attractor that represents the same EC behavior as that of their perturbed copies. Additionally, we calculated the sensitivity of each component of the update rule to molecular activation noise. For each update rule component $f_i \in f$, we first generated a set of 500,000 aleatory initial states. For each initial state, we created a perturbed copy with a Hamming distance of one by reversing the activatory state of one random variable. Then we applied the update rule once to each initial state and to its perturbed copy. The fraction of initial states, where after update the activatory state of the variable $x_i(t+1)$ is different for the initial state then it is for its perturbed copy is our estimation of the sensitivity for update rule f_i.

3. RESULTS

3.1. The Network Model

The model includes 143 nodes and 256 edges (**Figure 2**) the update rules of the network are included in Supplementary Section 2. To enable a more thorough analysis of the dynamic behavior of our model, we simplified our model and obtained a network composed of 64 nodes and 163 interactions, a diagram of our simplified model is shown in **Figure 3**. The update rules that define the dynamic behavior of our model are included as Equations 1–64. The EC behavior transitions integrated into both our detailed and simplified models are summarized in **Figure 4A**, and Supplementary Figures 1–14. Single gain- and loss-of-function mutations have a similar effect on the behaviors and transitions integrated into both models (Supplementary Figures 15, 16).

$$AKT(t+1) = PIP3(t) \tag{1}$$

$$ALK1(t+1) = BMP9(t) \vee BMP10(t) \vee TGFB1(t) \tag{2}$$

$$ALK5(t+1) = BMP9(t) \tag{3}$$

$$AMPATP(t+1) = AMPATP(t) \tag{4}$$

$$AMPK(t+1) = (AMPATP(t) \vee (\neg Oxygen(t))) \wedge (\neg AKT(t)) \tag{5}$$

$$ANG1(t+1) = ANG1(t) \tag{6}$$

$$ANG2(t+1) = (\neg KLF2(t)) \wedge (HIF1(t) \vee ETS(t) \vee AP1(t) \vee FOXO1(t)) \tag{7}$$

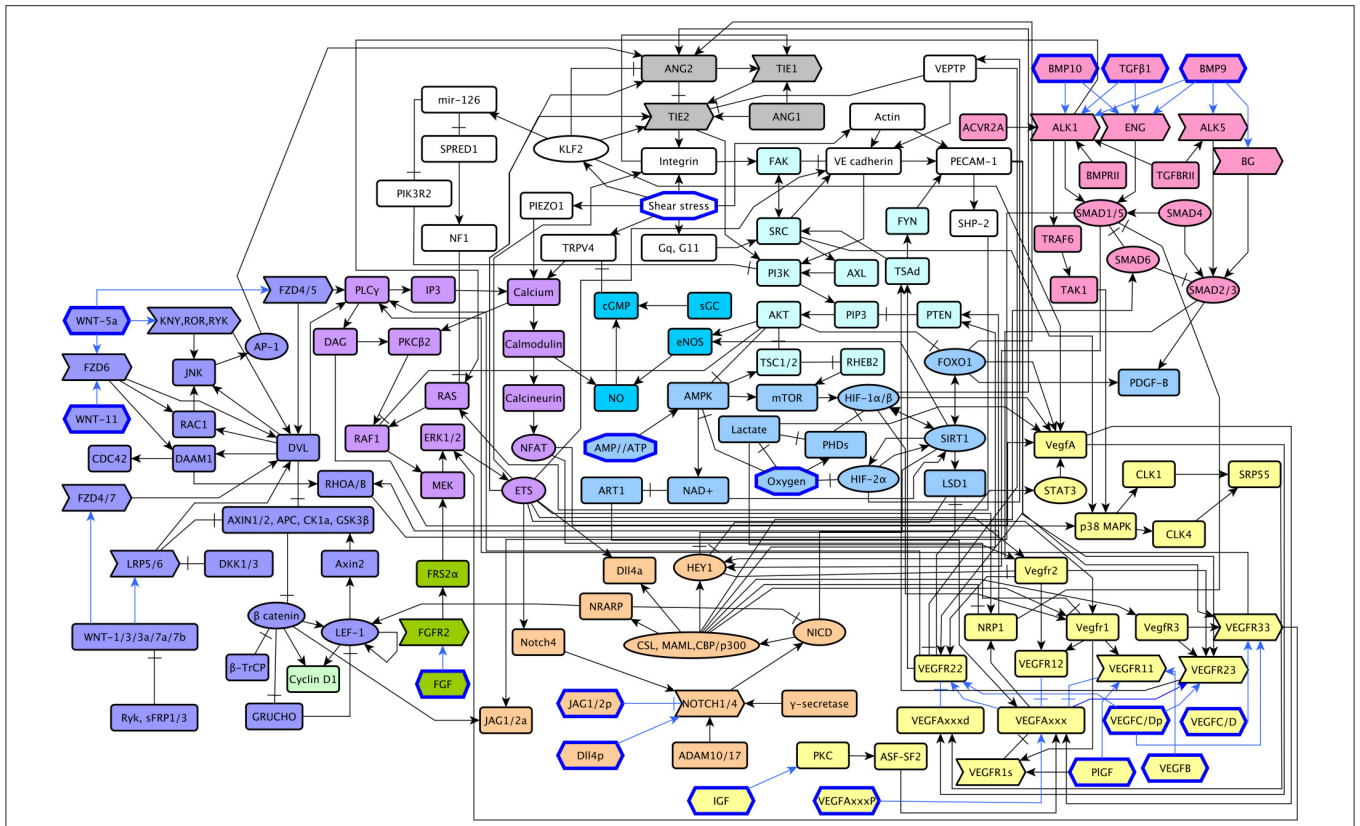


FIGURE 2 | A diagram of our extended model: The *ANG/TIE* signaling pathway is shown in gray, *Shear Stress* in white, *Oxygen and Energy* in blue, *NO* in turquoise, *VEGF* in yellow, *AKT/SRC* in light blue, *TGF* in pink, *NOTCH* in orange, *WNT* in purple, *RAS/PLCγ* in violet, *CyclinD1* in light green, and *FGF* in green. Ligands are represented as hexagons, other micro-environment variables as octagons, receptors as right arrows, transcription factors as ellipses, and signal transducers as rounded rectangles. Intracellular signaling is represented in black arrows, extracellular signaling is represented with blue arrows. Activatory interactions are shown as regular arrows and inhibitory interactions are shown as blunt arrows.

$$AP1(t+1) = WNT5a(t) \quad (8)$$

$$\beta catenin(t+1) = WNT5a(t) \vee WNT7a(t) \quad (9)$$

$$BMP10(t+1) = BMP10(t) \quad (10)$$

$$BMP9(t+1) = BMP9(t) \quad (11)$$

$$Calcium(t+1) = PLCg(t) \vee ShearStress(t) \vee (\neg NO(t)) \quad (12)$$

$$DLL4a(t+1) = ETS(t) \vee NICD(t) \quad (13)$$

$$DLL4p(t+1) = DLL4p(t) \quad (14)$$

$$ETS(t+1) = MEK(t) \vee VEGFR33(t) \quad (15)$$

$$FAK(t+1) = SRC(t) \vee Integrin(t) \quad (16)$$

$$FGF(t+1) = FGF(t) \quad (17)$$

$$FOXO1(t+1) = (\neg AKT) \wedge SIRT1(t) \quad (18)$$

$$HEY1(t+1) = NICD(t) \vee ((SMAD1(t) \vee SMAD2(t)) \wedge (\neg SIRT1(t))) \quad (19)$$

$$HIF1(t+1) = (AMPK(t) \vee \neg TSC(t)) \wedge \neg Oxygen(t) \wedge SIRT1(t) \quad (20)$$

$$IGF(t+1) = IGF(t) \quad (21)$$

$$Integrin(t+1) = ETS(t) \wedge (ShearStress(t) \vee TIE2(t)) \quad (22)$$

$$JAGa(t+1) = SMAD1(t) \vee \beta catenin(t) \quad (23)$$

$$JAGp(t+1) = JAGp(t) \quad (24)$$

$$KLF2(t+1) = ShearStress(t) \quad (25)$$

$$LEF1(t+1) = \beta catenin(t) \wedge (LEF1(t) \vee NRARP(t)) \quad (26)$$

$$MEK(t+1) = (((PLCg(t) \wedge Calcium(t)) \vee RAS(t)) \wedge (\neg AKT(t))) \vee FGF(t) \quad (27)$$

$$NFAT(t+1) = Calcium(t) \quad (28)$$

$$NICD(t+1) = (\neg NRARP(t)) \wedge NOTCH(t) \quad (29)$$

$$NO(t+1) = Calcium(t) \vee AKT(t) \vee SIRT1(t) \quad (30)$$

$$NOTCH(t+1) = (\neg JAGp(t)) \wedge ETS(t) \wedge DLL4p(t) \quad (31)$$

$$NRP1(t+1) = (VEGFAxxx(t) \vee VEGFC\_Dp(t)) \wedge (\neg NICD(t) \vee ETS(t)) \quad (32)$$

$$NRARP(t+1) = NICD(t) \quad (33)$$

$$Oxygen(t+1) = Oxygen(t) \quad (34)$$

$$p38MAPK(t+1) = SRC(t) \vee PLCg(t) \vee ALK1(t) \quad (35)$$

$$PECAM1(t+1) = VEGFR22(t) \vee VEGFR23(t) \vee ShearStress(t) \vee VEcadherin(t) \quad (36)$$

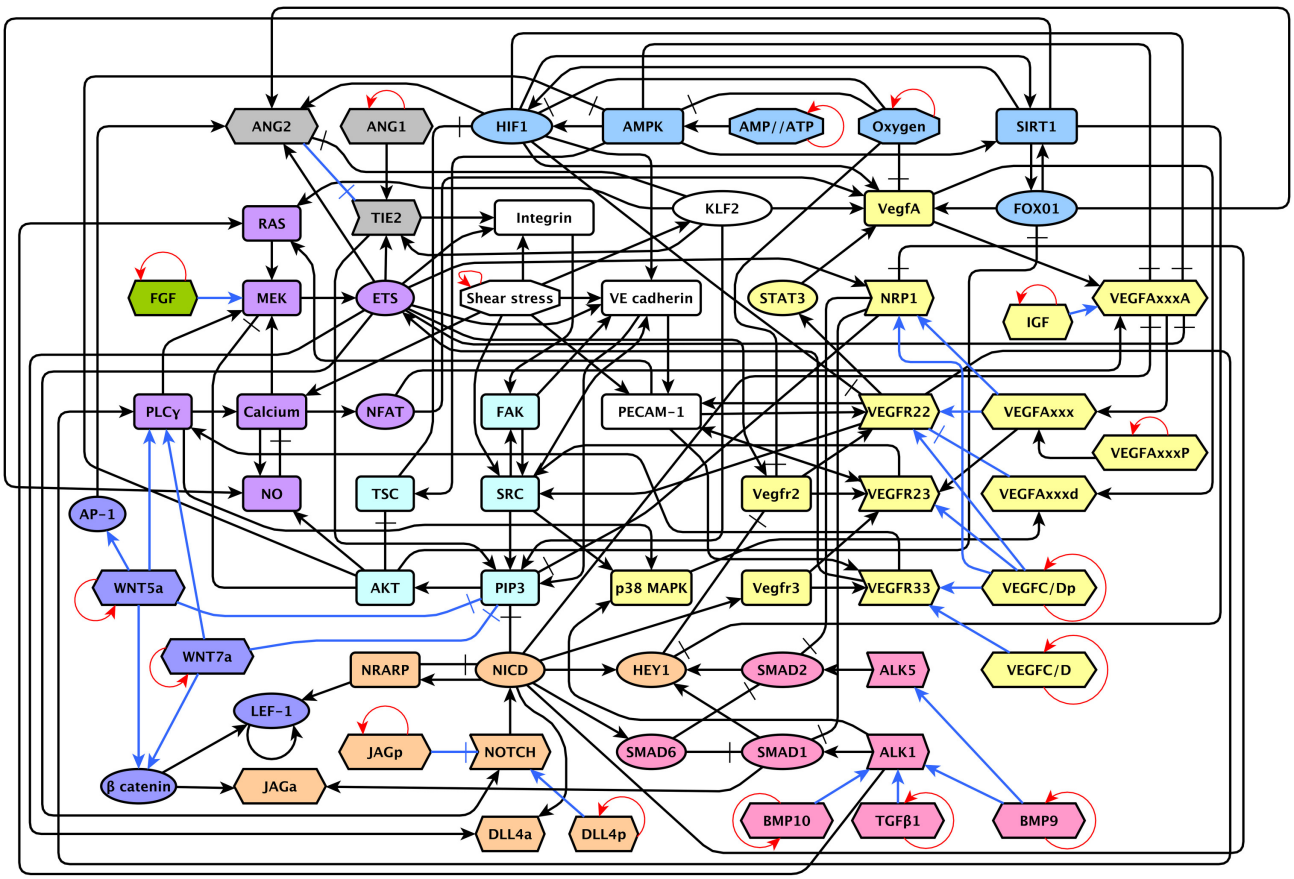


FIGURE 3 | A diagram of our reduced network model: The *ANG/TIE* signaling pathway is shown in gray, *Shear Stress* in white, *Oxygen and Energy* in blue, *VEGF* in yellow, *AKT/SRC* in light blue, *TGF* in pink, *NOTCH* in orange, *WNT* in purple, *RAS/PLCγ* in violet, and *FGF* in green. Ligands are represented as hexagons, other micro-environment variables as octagons, receptors as right arrows, transcription factors as ellipses, and signal transducers as rounded rectangles. Intracellular signaling is represented in black arrows, extracellular signaling is represented with blue arrows. Activatory interactions are shown as regular arrows and inhibitory interactions are shown as blunt arrows. The self activatory feedback circuits required to keep the micro-environment constant during the simulation are shown in red.

$$PIP3(t+1) = (\neg NICD(t)) \wedge (\neg WNT5a(t)) \wedge (\neg WNT7a(t)) \wedge (\neg NRP1(t)) \wedge (SRC(t) \vee KLF2(t) \vee VEcadherin(t) \vee TIE2(t)) \quad (37)$$

$$PLCg(t+1) = VEGFR22(t) \vee VEGFR33(t) \vee WNT5a(t) \vee WNT7a(t) \quad (38)$$

$$RAS(t+1) = PECAM1(t) \vee KLF2(t) \vee ALK1(t) \quad (39)$$

$$ShearStress(t+1) = ShearStress(t) \quad (40)$$

$$SIRT1(t+1) = AMPK(t) \wedge (HIF1(t) \vee FOXO1(t)) \quad (41)$$

$$SMAD1(t+1) = (\neg SMAD6(t)) \wedge (\neg NRP1(t)) \wedge ALK1(t) \quad (42)$$

$$SMAD2(t+1) = (\neg SMAD6(t)) \wedge (\neg NRP1(t)) \wedge ALK5(t) \quad (43)$$

$$SMAD6(t+1) = NICD(t) \quad (44)$$

$$SRC(t+1) = FAK(t) \vee ShearStress(t) \vee VEGFR22(t) \vee VEGFR23(t) \quad (45)$$

$$STAT3(t+1) = VEGFR22(t) \quad (46)$$

$$TGFB1(t+1) = TGFB1(t) \quad (47)$$

$$TIE2(t+1) = (\neg ANG2(t)) \wedge ANG1(t) \wedge (ETS(t) \vee KLF2(t)) \quad (48)$$

$$TSC(t+1) = AMPK(t) \wedge (\neg AKT(t)) \quad (49)$$

$$VEcadherin(t+1) = ETS(t) \wedge (SRC(t) \vee FAK(t) \vee ShearStress(t) \vee HIF1(t)) \quad (50)$$

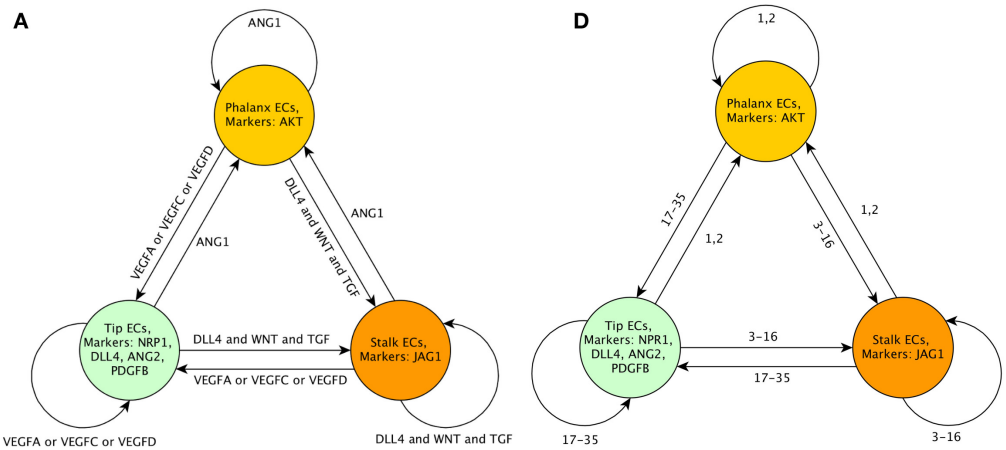
$$VegfA(t+1) = (\neg Oxygen(t)) \vee HIF1(t) \vee STAT3(t) \vee FOXO1(t) \vee NFAT(t) \vee KLF2(t) \quad (51)$$

$$VEGFAxxx(t+1) = VEGFAxxxP(t) \vee VEGFAxxxA(t) \quad (52)$$

$$VEGFAxxxA(t+1) = VegfA(t) \wedge IGF(t) \wedge ((\neg NICD(t) \wedge \neg HIF1(t) \wedge \neg ETS(t)) \vee NFAT(t)) \wedge (\neg AMPK(t)) \quad (53)$$

$$VEGFAxxxd(t+1) = p38MAPK(t) \wedge VegfA(t) \quad (54)$$

$$VEGFAxxxP(t+1) = VEGFAxxxP(t) \quad (55)$$



B

Group	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
VEGFC_Dp	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
VEGFAxxxP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	1	1	1	1	1	1	1	0	0	0	0	0
ANG1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Oxygen	*	*	0	0	1	1	0	0	1	1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	1	1	1	1	1
ShearStress	1	1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	1	*	*	*	*	*	*	1	*	*	1	0	0
JAGp	0	1	*	*	*	*	*	*	*	*	0	0	0	1	1	1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0	1
DLL4p	0	*	*	*	*	*	*	*	*	*	0	0	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0	*
WNT5a	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	*	*	*	*	*	1	0	*	*	*	*	*	1	0	*	1	0	0	0
WNT7a	0	0	1	*	1	*	1	*	1	*	0	0	0	0	0	0	*	*	*	*	1	0	0	*	*	*	*	1	0	0	1	0	0	0	0
FGF	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	1	0	0	0	*	*	*	1	0	0	0	*	*	*	1	1
IGF	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	1	1	1	1	1
BMP9	0	0	*	*	*	*	*	*	*	*	*	*	1	*	*	1	*	*	1	0	0	0	0	*	*	1	0	0	0	0	*	*	*	0	0
BMP10	0	0	*	*	*	*	*	*	*	*	*	1	0	*	1	0	*	1	0	0	0	0	0	*	1	0	0	0	0	0	*	*	*	0	0
TGFB1	0	0	*	*	*	*	*	*	*	*	1	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	*	*	*	0	0
VEGFC_D	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
AMPATP	*	*	*	*	*	*	*	*	1	1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0	0	0	0	0
log2(N.E)	5	6	10	11	10	11	10	11	9	10	8	7	6	9	8	7	14	13	12	11	10	9	8	13	12	11	10	9	8	7	10	9	8	2	3
Subtotal	96		9216						1536		1344						48768														1792			12	
Total	96		12096														50572																		
Behavior	Phalanx		Stalk														Tip																		

C

Group	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67
VEGFC_Dp	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
VEGFAxxxP	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ANG1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Oxygen	*	*	*	*	*	*	*	*	*	0	0	0	*	*	*	*	*	*	0	0	0	1	1	1	*	*	*	0	*	*	0	1
ShearStress	0	0	0	*	*	*	0	0	0	1	1	1	*	*	*	0	0	0	1	1	1	1	1	1	0	1	1	1	0	0	0	0
JAGp	*	*	*	*	*	*	*	*	*	*	*	*	0	0	0	0	0	0	0	0	0	0	0	0	*	0	*	*	*	*	*	0
DLL4p	*	*	*	*	*	*	*	*	*	*	*	*	1	1	1	1	1	1	1	1	1	1	1	1	*	1	*	*	*	*	*	1
WNT5a	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
WNT7a	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FGF	0	0	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0	1	1
IGF	*	*	*	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
BMP9	0	0	0	*	*	1	*	*	1	*	*	1	*	*	1	*	*	1	*	*	1	*	*	1	0	0	0	0	0	0	0	0
BMP10	0	0	0	*	1	0	*	1	0	*	1	0	*	1	0	*	1	0	*	1	0	*	1	0	0	0	0	0	0	0	0	0
TGFB1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
VEGFC_D	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
AMPATP	*	*	*	1	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	*	*	1	0	1	0	0	0
log2(N.E)	7	7	7	9	8	7	8	7	6	7	6	5	7	6	5	6	5	4	5	4	3	5	4	3	7	5	6	5	6	5	4	2
Subtotal	384			1568									392									56			128	32	96		112			4
Total	2772																															

FIGURE 4 | Endothelial cell behavior: Phalanx EC behavior shown in yellow, Stalk EC behavior shown in orange, Tip EC behavior shown in green, and other EC behavior is shown in gray: **(A)** Expected EC behavior in an extracellular micro-environment with normal oxygen concentration, ATP to ADP ratio and shear stress, **(B)** The extracellular micro-environments that cause Phalanx, Stalk, and Tip EC behavior according to the simulation of the dynamic behavior of our simplified model, **(C)** The extracellular micro-environments that cause other EC behavior, **(D)** Summary of EC behavior according to our model, the numbers on the edges represent the groups of micro-environments shown as columns in panel **(B)**.

$$VEGFC\_D(t+1) = VEGFC\_D(t) \quad (56)$$

$$VEGFC\_Dp(t+1) = VEGFC\_Dp(t) \quad (57)$$

$$Vegfr2(t+1) = (ETS(t) \wedge (\neg HEY1(t))) \vee \neg Oxygen(t) \quad (58)$$

$$VEGFR22(t+1) = Vegfr2(t) \wedge (PECAM1(t) \vee ((VEGFC\_Dp(t) \vee VEGFAxxx(t)) \wedge \neg(VEGFAxxxd(t) \vee HIF1(t))) \quad (59)$$

$$VEGFR23(t+1) = Vegfr2(t) \wedge Vegfr3(t) \wedge (PECAM1(t) \vee VEGFAxxx(t) \vee VEGFC\_Dp(t)) \quad (60)$$

$$Vegfr3(t+1) = NICD(t) \quad (61)$$

$$VEGFR33(t+1) = Vegfr3(t) \wedge (PECAM1(t) \vee VEGFC\_D(t) \vee VEGFC\_Dp(t)) \quad (62)$$

$$WNT5a(t+1) = WNT5a(t) \quad (63)$$

$$WNT7a(t+1) = WNT7a(t) \quad (64)$$

3.2. The Effect of the Extracellular Micro-environment on EC Behavior

One of the main goals of this work is to understand how the concentration of several molecules in the extracellular micro-environment combines with the mechanical forces sensed by the mechano-receptors connected to the cytoskeleton of ECs controls EC behavior. We propose that the presence (1) or absence (0) of sufficient *VEGFC_Dp, VEGFAxxxP, ANG1, Oxygen, ShearStress, JAGp, DLL4p, WNT5a, WNT7a, FGF, IGF, BMP9, BMP10, TGFB1, VEGFC_D*, and *AMPATP* in the micro-environment of an EC determines its behavior. Further, we propose that Phalanx, Stalk, and Tip ECs retain the ability respond to the micro-environment in a similar manner and that explains the plasticity in EC behavior that has been experimentally observed (Blancas et al., 2012; Glaser et al., 2016). To investigate the effect of the extracellular micro-environment on EC behavior, we first found all the attractors that can be reached through the simulation of the dynamic behavior of our model. Then, we classified them according to their extracellular micro-environment. After that we interpreted the EC behavior represented by the attractors in each micro-environment. If all the attractors that correspond to a certain micro-environment represent the same kind of EC behavior, then we can state that the micro-environment causes that EC behavior. If most micro-environments cause either Tip, Stalk, or Phalanx EC behavior, then to a large extent the extracellular micro-environment controls EC behavior.

Notably, there are $2^{16} = 65536$ possible micro-environments. From these, according to our model, under wild-type conditions 50,572 (77.16675%) micro-environments cause Tip EC behavior, 12,096 (18.45703%) cause Stalk EC behavior, and 96 (0.1464844%) cause Phalanx EC behavior. The characteristics of the groups of micro-environments that lead to Phalanx, Stalk, and Tip EC behavior are summarized in **Figure 4B** and **Table 1**. The intracellular molecules that are active or inactive in all the patterns of molecular activation in each group are also summarized in **Table 1**. The other 2,772 micro-environments (4.229736%) cause atypical dynamical patterns, including attractors that cycle between the Tip, Stalk, and/or Phalanx EC behaviors. This means that 62,764 (95.770374%) of the micro-environments induce a certain EC behavior regardless of the internal pattern of molecular activation (**Figure 4D**). Therefore, according to our model, in most cases, the extracellular micro-environment controls EC behavior.

Tip ECs are localized at the leading edge of vessel sprouts forming numerous long dynamic filipodia. Additionally, Tip cells migrate toward angiogenic stimuli, do not contribute to lumen formation, and seldom divide. Tip ECs are characterized by expressing high levels of DLL4, CXCR4, ANG2, PDGFB, receptors for axon guidance cues, such as the Netrin receptor UNC5B, APLN, various proteases like uPAR and NRP1, (del Toro et al., 2010; Blancas et al., 2012). We use NRP1 activity as a Tip EC-specific marker, and also require DLL4 expression, because DLL4 directly inhibits neighboring cells from becoming Tip ECs. Additionally, AKT must be inactive in Tip ECs. It is known that an increase above a certain threshold on the concentration of VEGFA or proteolytically processed VEGFC or D in the micro-environment surrounding an EC triggers Tip EC behavior (sections 1.2 and 1.10 in the Supplementary Material). According to the simulated dynamic behavior of our model, the micro-environments that include VEGFAxxxP or VEGFC_Dp and induce Tip EC behavior, also include either *ShearStress, WNT5a, WNT7a, FGF, BMP9, BMP10*, or *TGFB1*. Alternatively, the model also allows for the possibility that two groups of micro-environments that lack paracrine VEGF activity may cause Tip EC behavior, achieved by inducing autocrine VEGFA activity.

Stalk ECs trail Tip ECs, proliferate rapidly and contribute to lumen formation. While *TIE2* is constitutively expressed in all ECs, its protein is detectable by antibody staining on Stalk ECs but not on Tip ECs. Stalk cells also express the Apelin receptor *APJ* and *JAG1* (del Toro et al., 2010; Blancas et al., 2012). We use autocrine JAG1 as a Stalk EC marker due to the specificity of its expression and its function, which is to suppress Notch signaling in neighboring Tip ECs, further, Stalk ECs, are characterized by their lack of NRP1 activity. A sufficient concentration of WNT, TGF and NOTCH ligands, as well as an absence of VEGF in the extracellular micro-environment of an EC, is known to cause Stalk EC behavior (section 1.10 in the Supplementary Material). According to the simulated dynamic behavior of our model, it is possible to obtain the Stalk EC behavior in a micro-environment that complies with either of the following three lists of requirements: (a) WNT activity, lack of VEGF activity, and low Oxygen or IGF; (b) WNT activity, no VEGF activity, Oxygen, IGF, and sufficient energy; and (c) lack of VEGF, NOTCH, WNT, and IGF ligands that includes one of the TGF ligands.

Phalanx ECs form strong EC–EC bonds to compose the tunica intima in stable blood vessels. The Pericytes and SMCs that cover stable blood vessels secrete ANG1 to maintain the integrity of the layer of Phalanx ECs. Phalanx ECs are characterized by a high level of *VEGFR1 (FLT1)* and *TIE1* expression (Blancas et al., 2012), even though neither is a Phalanx EC specific marker. We use active AKT (Kerr et al., 2016) as well as inactive NRP1 and JAGa as specific Phalanx EC markers. Changes in the extracellular concentration of VEGFs, a decrease in the

TABLE 1 | Phalanx, Stalk, and Tip EC behavior: The groups correspond to those in **Figure 4B**, active molecules shown in blue, inactive molecules shown in red.

Behavior (groups, micro–environments, attractors)	Micro–environment characteristics	Molecular activity inside the cell
Phalanx (1–2, 96, 96)	ShearStress, and VEGFC_Dp, VEGFAxxxP, WNT5a, WNT7a, IGF, BMP9, BMP10, TGFB1, and (JAGp or DLL4p)	RAS, KLF2, VegfA, Calcium, NFAT, FAK, PECAM1, NO, SRC, VEGFAxxxd, AKT, PIP3, p38MAPK, and ANG2, HIF1, AMPK, SIRT1, FOXO1, NRP1, VEGFAxxxA, VEGFAxxx, TSC, VEGFR23, AP1, Vegfr3, VEGFR33, catenin, LEF1, NRARP, NICD, HEY1, SMAD2, ALK5, JAGa, NOTCH, SMAD6, SMAD1, ALK1 do not divide or recruit mural cells.
Stalk I (3–8, 9216, 58896)	VEGFC_Dp and VEGFAxxxP and (WNT5a or WNT7a), and (Oxygen or IGF)	VegfA, MEK, ETS, PLCg, Calcium, NFAT, NO, VEGFAxxd, p38MAPK, βcatenin, JAGa, DLL4a, NRP1, VEGFAxxxA, VEGFAxxx, AKT, PIP3
Stalk II (9–10, 1536, 14688)	Oxygen and IGF and AMPATP and VEGFC_Dp and VEGFAxxxP and (WNT5a or WNT7a)	AMPK, Oxygen, VegfA, AMPATP, MEK, ETS, IGF, PLCg, Calcium, NFAT, NO, TSC, VEGFAxxd, p38MAPK, βcatenin, JAGa, DLL4a NRP1, VEGFAxxxA, VEGFAxxx, AKT, PIP3
Stalk III (11–16, 1344, 3276)	VEGFC_Dp and VEGFAxxxP and WNT5a WNT7a and IGF and (JAGp or DLL4p) and (BMP9 or BMP10 or TGF1)	RAS, p38MAPK, JAGa, SMAD1, ALK1, HIF1, NRP1, IGF, VEGFAxxxA, VEGFAxxx, VEGFR23, βcatenin, LEF1, NICD, NOTCH, SMAD6 do not divide
Tip I (17–30, 48768, 244680)	(VEGFC Dp, or VEGFAxxxP) and (ShearStress or WNT5a or WNT7a or FGF or BMP9 or BMP10 or TGFB1)	VegfA, MEK, ETS, NRP1, VEGFAxxxd, p38MAPK, DLL4a, AKT, PIP3, SMAD1, SMAD2
Tip II (31–33, 1792, 4096)	VEGFAxxxP and VEGFC Dp and AMPATP and IGF and Oxygen and (ShearStress or WNT5a or WNT7a)	Oxygen, VegfA, MEK, ETS, NRP1, Calcium, NFAT, VEGFAxxxA, VEGFAxxx, NO, VEGFAxxxd, p38MAPK, DLL4a, HIF1, AMPK, SIRT1, FOXO1, TSC, AKT, PIP3, SMAD1, SMAD2 do not recruit mural cells
Tip III (34–35, 12, 12)	VEGFAxxxP and VEGFC Dp and AMPATP and IGF and Oxygen and ShearStress and WNT5a and BMP9 and BMP10 and TGF1 and WNT7a and (JAGp or DLL4p)	ANG2,Oxygen, RAS, VegfA, FGF, MEK, ETS, VEcadherin, STAT3, NRP1, PLCg, Calcium, NFAT, FAK, PECAM1, VEGFR22, VEGFAxxxA, VEGFAxxx, NO, SRC, Vegfr2, VEGFAxxxd, p38MAPK, DLL4a, TIE2, HIF1, AMPK, SIRT1, Integrin, KLF2, FOXO1, TSC, VEGFR23, AP1, AKT, PIP3, Vegfr3, VEGFR33, βcatenin, LEF1, NRARP, NICD, HEY1, SMAD2, ALK5, JAGa, NOTCH, SMAD6, SMAD1, ALK1 do not divide and do not recruit mural cells

βcatenin and LEF1 activity is required to allow Cyclin D1–mediated activation of the cell cycle. FOXO1 or SMAD2 activity is required for PDGFβ–mediated mural cell recruitment.

availability of oxygen or energy within the cell, and shear stress cause ANG2-mediated activation of the ECs that line blood vessels (section 1.8 in the Supplementary Material). According to our model, the lack of *VEGF, NOTCH, WNT* and *TGF* pathway activity is necessary to observe a stable Phalanx EC behavior. The simulated Phalanx ECs do not divide and do not recruit mural cells.

3.2.1. Atypical EC Behavior

We performed with our model a systematic study of the dynamical behavior of a regulatory network under all possible combination of the micro-environments. Apart from the clearly identifiable phenotypes mentioned in the previous paragraphs, we observed some atypical responses. If the attractors that correspond to a certain micro-environment represented different EC behaviors, or any of the attractors represented an EC behavior that was different from Tip, Stalk, or Phalanx EC behavior, we considered that the micro-environment causes atypical EC behavior. For completeness, we describe such atypical behaviors in **Table 2**.

3.2.2. EC Proliferation

EC proliferation allows the number of ECs to increase during sprout elongation. We describe the effect of the micro-environment on EC proliferation according to the simulated dynamic behavior of our model in **Table 3**. Note that in accordance with what has been reported in the literature, only Tip and Stalk ECs proliferate.

3.3. Model Validation

Certain diseases exhibit abnormal angiogenesis, because the affected tissue or organ secretes abnormal amounts of angiogenic signals. Simulating the effect of a pathological extracellular micro-environment on EC behavior can be used to understand how a disease is causing abnormal vascular remodeling, the insights are only valid if the dynamic behavior of the model can reproduce the relevant experimental observations. If an experimental observation includes a sufficiently well-defined extracellular micro-environment and an observed EC behavior. Then the extracellular micro-environment fits only one column in **Figure 4B** or **Figure 4C**. If the EC behavior according to our model (shown at the bottom row of the column that corresponds to the micro-environment) is the same as the observed EC behavior, then our model fits that experimental observation.

3.3.1. Tumor Angiogenesis

The micro-environment inside many tumors is hypoxic, containing a high concentration of VEGFA and FGF. This state causes the formation of many leaky blood vessels (Nussenbaum and Herman, 2010). Our model can describe this state, as shown in **Figure 4B** group 27. The results indicate that the mentioned micro-environment induces Tip EC behavior, and inhibits

TABLE 2 | Atypical EC behavior: The groups correspond to those in **Figure 4C**, active molecules shown in blue, inactive molecules shown in red.

Behavior (groups, micro–environments, attractors)	Micro–environment characteristics	Molecular activity inside the cell
Atypical I (36–38, 384, 3920)	(VEGFAxxxP or VEGFC Dp) and ShearStress, and WNT5a, and WNT7a, and FGF, and BMP9, and BMP10, and TGFB1	KLF2, FGF, AP1, βcatenin, LEF1, SMAD2, JAGa, SMAD1,ALK1
Atypical II (39–47, 1568, 11172)	IGF and VEGFC_Dp and VEGFAxxxP and WNT5a and WNT7a and (AMPATP or Oxygen or ShearStress) and (BMP9 or BMP10 or TGFB1)	RAS, p38MAPK, ALK1, HIF1, AP1, βcatenin, LEF1
Atypical III (48–56, 392, 1876)	VEGFC_Dp and VEGFAxxxP and WNT5a and WNT7a and JAGp and IGF and DLL4p and (AMPATP or Oxygen or ShearStress) and (BMP9 or BMP10 or TGFB1)	RAS, p38MAPK, ALK1, HIF1, NRP1, VEGFAxxxA, VEGFAxxx, AP1, βcatenin and LEF1
Atypical IV (57–59, 56, 112)	VEGFC_Dp and VEGFAxxxP and WNT5a and WNT7a and JAGp and IGF and AMPATP and Oxygen and ShearStress and DLL4p and (BMP9 or BMP10 or TGFB1)	RAS, KLF2, VegfA, Calcium, NFAT, FAK, PECAM1, NO, SRC, VEGFAxxxd, p38MAPK, ALK1, ANG2, HIF1, SIRT1, FOXO1, NRP1, VEGFAxxxA, VEGFAxxx, TSC, AP1, βcatenin, LEF1
Atypical V (60, 128, 687)	VEGFC_Dp and VEGFAxxxP and ShearStress and WNT5a and WNT7a and IGF and BMP9 and BMP10 and TGFB1	KLF2, NRP1, VEGFAxxxA, VEGFAxxx, AP1, βcatenin, LEF1, SMAD2, ALK5, JAGa, ALK1
Atypical VI (61, 32, 64)	ShearStress and DLL4p and VEGFC_Dp and VEGFAxxxP and WNT5a and WNT7a and JAGp and IGF and BMP9 and BMP10 and TGFB1	RAS, KLF2, VegfA, Calcium, NFAT, FAK, PECAM1, NO, SRC, VEGFAxxxd, p38MAPK, ANG2, HIF1, SIRT1, FOXO1, NRP1, VEGFAxxxA, VEGFAxxx, AP1, βcatenin, LEF1, SMAD2, ALK5, JAGa, SMAD1, ALK1 do not recruit mural cells
Atypical VII (62–63, 96, 456)	ShearStress and IGF and VEGFC_Dp and VEGFAxxxP and WNT5a and WNT7a and BMP9 and BMP10 and TGFB1 and (AMPATP or Oxygen)	RAS, KLF2, VegfA, Calcium, NFAT, FAK, PECAM1, NO, SRC, VEGFAxxxd, p38MAPK, ANG2, HIF1, SIRT1, FOXO1, AP1, βcatenin, LEF1, SMAD2, ALK5, SMAD1, ALK1 do not recruit mural cells
Atypical VIII (64–66, 112, 1358)	IGF and VEGFC_Dp and VEGFAxxxP and ShearStress and WNT5a and WNT7a and BMP9 and BMP10 and TGF and (AMPATP or FGF or Oxygen)	KLF2, AP1, βcatenin, LEF1, SMAD2, ALK5, SMAD1 , ALK1
Atypical IX (67, 4, 8)	VEGFC_Dp and VEGFAxxxP and ShearStress and JAGp and WNT5a and WNT7a and BMP9 and BMP10 and TGF and AMPATP and Oxygen and DLL4p and FGF and IGF	ANG2, MEK, ETS, NOTCH, DLL4a, TIE2, HIF1, AMPK, SIRT1, Integrin, KLF2, FOXO1, TSC, AP1, βcatenin, LEF1, SMAD2, ALK5, JAGa, SMAD1, ALK1, DLL4a do not recruit mural cells

All the atypical EC behaviors include a quiescent cell cycle because βcatenin and LEF1 activity is required to allow Cyclin D1–mediated activation of the cell cycle. FOXO1 or SMAD2 activity is required for PDGFβ–mediated mural cell recruitment.

TABLE 3 | EC proliferation: Cyclin D1–mediated activation of the cell cycle requires βcatenin and LEF1 activity. Active molecules shown in blue, inactive molecules shown in red.

Behavior (micro–environments, attractors)	Micro–environment characteristics	Molecular activity inside the cell
All divide (12288, 42432)	JAGp and DLL4p and (WNT5a or WNT7a)	VegfA, MEK, ETS, PLCg, Calcium, NFAT, NO, VEGFAxxd, p38MAPK, βcatenin, LEF1, JAGa, NOTCH, DLL4a, AKT, PIP3
Some divide (36864, 244656)	(JAGp or DLL4p) and (WNT5a or WNT7a)	VegfA, MEK, ETS, PLCg, Calcium, NFAT, NO, VEGFAxxd, p38MAPK, βcatenin, JAGa, DLL4a, VEGFR23, AKT, PIP3, Vegfr3, VEGFR33, NRARP, NICD, NOTCH, SMAD6
None divide (16384, 58309)	WNT5a and WNT7a	βcatenin, LEF1, AP1

Phalanx EC behavior, which is consistent with experimental observations.

3.3.2. Pathological Ocular Angiogenesis

Diabetic retinopathy, age-related macular degeneration, retinopathy of prematurity, and other irreversible causes of blindness involve pathological angiogenesis. The capillaries of the retina are unique, the inner layer of the blood-retinal barrier is like that of other capillaries, and is composed of a single layer of ECs. However, the outer layer of the blood-retinal barrier is formed by retinal pigment epithelial cells instead of pericytes and SMCs. Pathological ocular angiogenesis is triggered by hypoxia from neuronal metabolism, inflammatory signals, and oxidative stress. Those micro-environmental conditions cause retinal pigmented epithelium, astrocytes, Müller cells, ECs, ganglion cells to secrete VEGFA (Siemerink et al., 2010). According to our model, the Tip ECs that secrete VEGFA during pathological ocular angiogenesis are likely exposed the extracellular micro-environments in groups 34–35 in **Figure 4B**, and are affected by oxidative stress, lack of shear stress and have sufficient oxygen. The other Tip ECs involved in pathological ocular angiogenesis and induced by paracrine VEGFA correspond to groups 17–30 in **Figure 4B**.

Other angiogenic pathologies are caused by mutations that affect how an EC responds to changes in the extracellular micro-environment. We used our simplified model to simulate the effect of all single gain- and loss-of-function mutations on EC behavior. Specifically, we analyzed how each mutation affects the groups of extracellular micro-environments that cause Tip, Stalk, and Phalanx EC behaviors in our simplified model. The effect of some of the mutations has been observed experimentally and it should be possible to simulate the observed behavior using our model. The expected effect of reducing, or enlarging the number of extracellular micro-environments that cause each EC behavior depends on the likelihood of appearance of each micro-environment. Only when almost all or none of the micro-environments lead to a certain EC behavior, and the mutation has been observed *in–vitro* or *in–vivo* it is possible to compare the simulated effect of a certain mutation (Supplementary Table 13) with its experimentally observed effect.

Simulated loss of autocrine function of *DLL4, ETS, MEK*, or *NRP1*, leads to the loss of functional Tip EC behavior, strongly favoring Stalk EC behavior. Importantly, all four mutations have been observed to cause severe vascular defects *in vivo* and *in vitro* (Supplementary Tables 6, 10, and 12). The loss of autocrine DLL4 leads to the formation of a higher number of Tip ECs that do not inhibit their neighbor ECs from becoming Tip ECs (del Toro et al., 2010).

Simulated gain-of-function mutations for proteolytically active VEGFA, VEGFC, and VEGFD as well as NRP1, prevent Stalk EC behavior and cause more than 99% of the extracellular micro-environments to induce Tip EC behavior. *In vivo* and *in vitro*, proteolytically active VEGFA, VEGFC, and VEGFD increase blood vessel branching, angiogenesis, and permeability (Supplementary Tables 11, 12).

Simulations indicate that the Phalanx EC behavior is prevented by a loss of AKT, PIP3, or ShearStress function, or alternatively by constitutive ALK1, βcatenin, BMP10, BMP9, IGF, autocrine JAG, NICD, NOTCH, NRP1, SMAD1, TGFβ1, proteolytically active VEGFA, VEGFC, or VEGFD, WNT5a, or WNT7a activity. *In vitro* and *in vivo*, loss of AKT, PIP3, or ShearStress leads to mural cell loss, blood vessel destabilization and regression (Supplementary Tables 2, 3). Constitutive βcatenin, IGF, NOTCH, NRP1, SMAD1, proteolytically active VEGFA, VEGFC, or VEGFD, WNT5a, or WNT7a activity induces EC migration, proliferation, survival, or angiogenesis (Supplementary Tables 4, 8–12).

3.4. Robustness Analysis

Molecular regulatory networks must balance the need to ignore noise perturbations with the need to respond adequately to stimuli. A Boolean network can be classified as ordered, critical, or chaotic. Ordered Boolean networks resist most perturbations without any important changes in their dynamic behavior and are not sufficiently sensitive to stimuli. Chaotic Boolean networks tend to magnify perturbations and do not resist enough noise. Critical Boolean networks are selectively sensitive to certain perturbations and are sufficiently resilient to noise to be adequate models of molecular regulatory networks (Lloyd-Price et al., 2012). Additionally, the robustness of each trait has specific implications.

3.4.1. The Robustness of Tip, Stalk, and Phalanx EC Behavior to Single Gain and Loss-of-Function Mutations

The resilience of a functional phenotype to changes in the genotype allows the accumulation of genetic variation in a population, and needs to be achieved without limiting excessively the ability of a species to adapt by evolving different traits (Kirschner and Gerhart, 1998; Jiménez et al., 2015). The simulations showed that $23/128 = 17.96875\%$ of all single gain- and loss-of-function mutations did not affect EC behavior at all. Furthermore, $82/128 = 64.0625\%$ of mutations only cause changes in the response of an EC to certain extracellular micro-environments. The other $23/128 = 17.96875\%$ of the mutations led to the loss of an EC behavior. Then, $4/128 = 3.125\%$ of all mutations cause loss of Tip EC behavior. The same number of mutations cause Stalk EC behavior loss and strongly favor Tip EC behavior. Finally, $18/128 = 14.0625\%$ of the mutations cause loss of Phalanx EC behavior. This set of results imply that our model of the network is robust to the complete loss of any of the main EC behaviors, however many mutations change the number of micro–environments that cause Tip, Stalk, and Phalanx EC behaviors (Supplementary Tables 13, 14).

3.4.2. The Robustness of Attractor Determination and EC Behavior to Molecular Activation Noise

Only 33.0538% of the trajectories followed by the perturbed copies of 1,000,000 random initial states reached the same attractor as the original state. In contrast, when we used 1,000,000 random initial states to test the robustness of EC behavior to molecular activation noise in 98.90088% of the relevant experiments the perturbation did not affect Tip EC, in 95.30536% of the relevant experiments, the perturbation did not affect Stalk EC behavior, and in 86.58824% of the relevant experiments, the perturbation did not affect Phalanx EC behavior. In general, 97.91060% of the random initial states reached the same EC behavior as the one reached by their perturbed copies.

3.4.3. The Sensitivity of Each Component of the Update Rule to Molecular Activation Noise

To understand which variables are more sensitive to stimuli and which ones tend to be more resilient to molecular activation noise. We estimated the sensitivity of each component of the update rule as described in the methods section. The results are shown in **Figure 5** and the sensitivity values in section 2.1 in the Supplementary Material. The nodes with the six most sensitive update rules in our network are NRP1, MEK, Integrin, HEY1, SIRT1, AMPK. Even the update rule of NRP, the most sensitive in our model, has a relatively low sensitivity of 0.023716.

4. DISCUSSION

We presented in this work a reconstruction of the regulatory network involved in the control of angiogenesis, integrating the largest set of canonical signaling pathways to date.

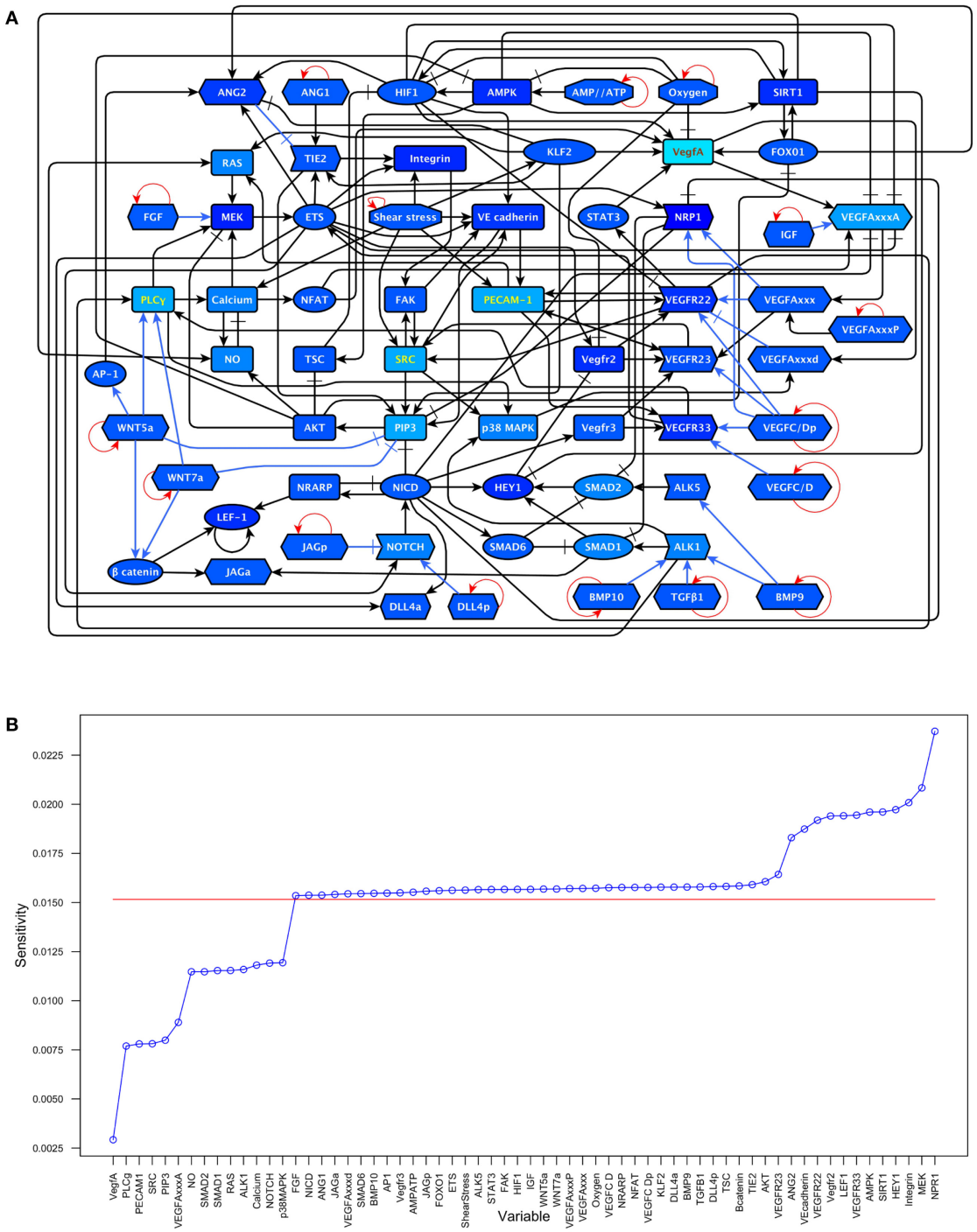


FIGURE 5 | Update rule component sensitivity: **(A)** A darker shade of blue indicates a higher sensitivity in the update rule. Values range from VegfA = 0.002926 to NRP1 = 0.023716. **(B)** The sensitivities of the components of the update rule arranged from smallest to largest compared to the average sensitivity (0.01515947) which is shown as a red line.

The dynamical behavior of the network, simulated as a Boolean network model, recovered the qualitative patterns of molecular activation observed in Phalanx, Tip, and Stalk ECs. Furthermore, the simulated behavior of the model corresponded to what has been reported in the literature regarding the high degree of behavioral plasticity between Phalanx, Stalk, and Tip EC behaviors in response to specific molecular micro-environments. Moreover, the model was also

able to describe the effect of gain- and loss-of-function mutations.

4.1. Insights and Predictions Based on the Simulated Dynamic Behavior of Our Model

The qualitative agreement between our model and published data shows that the model is a useful framework to understand the mechanisms that underly normal angiogenesis. Furthermore, it allows generating hypotheses on the mechanisms by which a disruption in the system might lead to deviation in EC behavior, which might eventually lead to a pathogenic phenotype. The qualitative agreement between our model and published results cannot be attributed to some sort of model fitting. This is evidenced by the high robustness observed in the model against the complete loss of any of the main EC behaviors (Supplementary Tables 13, 14), despite the perturbations introduced in the update rules. Nonetheless, when we analyzed the effect of single gain- and loss-of-function mutations, the simulations recovered the observed effects of such mutations under certain micro-environments.

Our micro-environment EC behavior map allows us to put forward the following hypotheses about the requirements for Tip Stalk and Phalanx EC behaviors: (1) In a micro-environment with an active, paracrine VEGF ligand, the presence of either *ShearStress, WNT5a, WNT7a, FGF, BMP9, BMP10*, or *TGFB1* is necessary to induce Tip EC behavior. (2) A micro-environment without VEGF can induce Tip EC behavior if it includes Oxygen, nutrients and IGF (Tip II, and Tip III in **Table 1**). However, the resulting Tip ECs secrete autocrine VEGFA. (3) DLL4 is not required for a micro-environment to induce Stalk EC behavior. (4) Shear stress and the absence of VEGF, TGF, IGF, WNT, and NOTCH ligands in the micro-environment is needed to observe a stable Phalanx EC behavior.

Based on the simulated effect of constitutive NRP1 activity, we predict that it prevents Stalk EC behavior and induces Tip EC behavior. We predict that constitutive ALK1, BMP9, BMP10, autocrine JAG, NICD, NRP1, SMAD1, and TGFβ1 activity inhibits Phalanx EC behavior based on the simulated effect of the corresponding gain-of-function mutations. Therefore, the model helps predict which mutations cause augmented mural cell loss, EC migration, proliferation, and angiogenesis, concomitant with inhibited Phalanx EC behavior.

Knowing the response of endothelial cells under a specific micro-environment is extremely relevant because inhibiting angiogenesis is an important medical goal during the treatment of vascular retinal disorders and cancer. Most of the drugs that are used to inhibit angiogenesis target the VEGF signaling pathway, inhibiting Tip EC behavior (Yadav et al., 2015). Our model suggests alternative ways to eliminate Tip EC behavior. Specifically, by eliminating the function of DLL4, ETS, MEK, or NRP1. Notably, both NRP1 and DLL4 are located on the cell membrane of ECs and are therefore easily reachable by drugs. Furthermore, in vascular retinal disorders, vascular permeability increases and vascular integrity diminishes, that is associated with intra-ocular hemorrhage and invasive potential of cancer. In principle, an extracellular micro-environment conducive to Phalanx EC behavior would help increase vascular integrity. Finally, stimulating angiogenesis is also an important medical goal during wound healing. It would be possible, thus, to use our model to explore one of the micro-environments that lead to Tip EC behavior and therefore, induce the wound healing process.

Arteriovenous malformations are very frequent in patients who suffer from Hereditary Hemorrhagic Telangiectasia (HHT), a disease associated with reduced ALK1, ENG, or SMAD4 function. In addition, Pulmonary Arterial Hypertension (PAH) is associated with reduced BMPRII or SMAD1 function. Furthermore, venous malformations have been observed in mice with constitutive TIE2 activity, as well as in mice with loss of ERK function. According to our model, the simulated effect of the mutations mentioned above includes an increase in the number of micro-environments that lead to Phalanx EC behavior, suggesting that the mentioned diseases are a consequence of ectopic blood vessel stabilization.

4.2. Assumptions and Limitations of Our Model

In this first version of the model of angiogenesis, we focus on the effect of the extracellular micro-environment on the behavior of a single endothelial cell. By using a Boolean model, we assume that all variables can only be active or inactive. Further, we use a synchronous update approach, therefore, we assume that all variables are activated or inhibited simultaneously. The limitations of our model affect the number of sprouting angiogenesis processes that we can reproduce and the extent to which we can simulate them. Some of the processes that are beyond the scope of our model have been studied using other previously published models (Peirce, 2008; Qutub et al., 2009; Scianna et al., 2013; Logsdon et al., 2014; Heck et al., 2015; Qutub and Popel, 2015) while other processes offer opportunities for further research as specified in the following paragraphs.

4.2.1. Secretion of Angiogenic Factors

According to our model, certain conditions cause ECs to secrete vascular growth factors (**Figure 4B** columns 31–35), the conditions that cause ECs to secrete active VEGFA (VEGFAxxxA) include sufficient oxygen, IGF, and a low AMP to ATP ratio. Normally, ECs are in contact with blood preventing hypoxia and lack of nutrients. The cells that compose other tissues respond to hypoxia or a high AMP to ATP ratio by secreting angiogenic factors; however, those cells are not included in our model. Additionally, Oxygen and then the secreted VEGF form concentration gradients. A continuous model that includes the geometry of the region or organ of interest as a boundary condition is necessary to simulate the gradient. Moreover, VEGFR1s secretion modulates the VEGFA concentration gradient (Chappell et al., 2016).

4.2.2. Vessel Destabilization

ANG2 activity is associated with mural cell detachment and it is possible to reproduce EC behavior during blood vessel destabilization using our model. However, it is not possible to reproduce pericyte and smooth muscle cell detachment because they are not included in our model. Some previous modeling efforts have included blood vessel destabilization (Zheng et al., 2013). However, in our opinion, mural cell behavior during angiogenesis merits a more detailed exploration.

4.2.3. Tip and Stalk Cell Differentiation

We carefully analyzed tip and stalk EC differentiation using our model emphasizing the interaction between the VEGF, WNT, TGF, NOTCH, Calcium, and NO signaling pathways during Tip and Stalk behavior specification. It is noteworthy that while Tip cells induce Stalk behavior in their neighbors by expressing DLL4 (Blanco and Gerhardt, 2013), according to our model NOTCH signaling inhibits Tip EC behavior only in a small group of micro-environments (**Figure 4B**, columns 34 and 35). A possible explanation for this apparent discrepancy is that active NOTCH signaling induces the secretion of VEGFR1s, which binds VEGFA, effectively raising the extracellular concentration of VEGFA needed to induce Tip EC behavior in the cells with active NOTCH signaling. In our Boolean model, it is not possible to include the changing VEGFAxxxP threshold, this would require a continuous model. Further, at the multicellular level, the chronological order in which ECs are affected by VEGFA and DLL4-mediated lateral inhibition creates a race condition (Bentley and Chakravartula, 2017). The temporal modulation of Tip and stalk EC behavior, including the effect of filipodia on tip cell sensitivity to VEGF, has been explored by previous modeling efforts (Venkatraman et al., 2016). A continuous, asynchronous, multicellular model that includes Matrix metalloproteinase, Apelin signaling (Palm et al., 2016) and VEGFR1s secretion (Chappell et al., 2016) would offer additional valuable insights.

4.2.4. Sprout Elongation

We simulated the micro-environmental conditions that may cause ECs to divide. However, our model does not include cell shape, which also changes during sprout elongation. Further sprout elongation is a multicellular process and our model includes only one EC. Several previous modeling efforts have studied sprout elongation (Logsdon et al., 2014). The authors of Norton and Popel (2016) analyzed the effect of EC proliferation, elongation, and migration during sprout elongation. Mechanical forces regulate both the location of sprout initiation and the rate of sprout elongation (Ghaffari et al., 2015), included in the model proposed by the authors of Vavourakis et al. (2017). A multi-scale model including cytoskeletal dynamics, molecular activation, and mechanical forces would greatly enhance our understanding of sprout elongation.

4.2.5. Lumen Formation and Expansion

PIP3, FAK, and SRC activity has been associated with vacuole secretion that is one of the main processes involved in lumen formation. According to the simulated dynamic behavior of our model, all Phalanx cells secrete vacuoles, additionally, type III Stalk ECs may also secrete vacuoles. Lumen formation is a multicellular process, that involves vacuole secretion and cytoskeletal remodeling. Simulating lumen formation, EC repulsion and flow-mediated lumen formation is beyond the scope of our current model. The authors of Boas and Merks (2014) focused their modeling efforts on the study of lumen formation.

4.2.6. Anastomosis

Is a multicellular process that involves cytoskeletal remodeling including specific shape changes that are beyond the scope of our model. Anastomosis has been included in several 2D and 3D models (Zheng et al., 2013; Norton and Popel, 2016). ECs with a reduced concentration of membrane-localized VEGFR1 are more likely to form stable connections with incoming sprouts (Nesmith et al., 2017). A multicellular model that integrates VEGFR1 regulation, and how it affects anastomosis, may help explain micro–vascular architecture.

4.2.7. Vessel Stabilization

Phalanx EC behavior is expected in stable blood vessels and is recovered by our model. PDGFB-mediated mural cell recruitment is also recovered by our model. Other multicellular effects of vessel stabilization, such as decreased blood vessel permeability, are beyond the scope of our model. Some previous modeling efforts have included blood vessel stabilization (Zheng et al., 2013). However, in our opinion, mural cell behavior during angiogenesis merits a more detailed exploration.

4.2.8. Pruning

Some of the micro-environments that cause atypical EC behavior without VEGF, FGF, IGF, and without Shear Stress (**Figure 4C**, group 60) may correspond to EC behavior during pruning. However, pruning involves changes in EC shape, EC fusion events, and EC migration, which have not been included in our model. Pruning is mainly regulated by blood flow. Apoptosis is implicated in the regression of large diameter blood vessels. In the small-diameter blood vessels that are remodeled by angiogenesis, pruning involves EC migration, self-fusion, and contraction before reabsorption into the remaining vasculature (Korn and Augustin, 2015; Betz et al., 2016). The model proposed by the authors of Chen et al. (2012) provided valuable insights into the role of hemodynamics during Zebrafish midbrain vascular pruning.

In conclusion, we developed a Boolean model of the network involved in EC behavior control during angiogenesis. The simulated dynamic behavior of our model corresponds with what has been observed experimentally and published about EC behavior and the effect of single gain- and loss-of-function mutations. The dynamical behavior of the model can qualitatively describe a wide variety of physiopathological states during angiogenesis. We believe that this characteristic makes the model a good platform to study the effect of altering the micro-environments and/or molecular backgrounds on endothelial cells.

AUTHOR CONTRIBUTIONS

NW, LM, IG, and JK planned the research, wrote the article, analyzed, and discussed the results. NW reviewed the literature, composed the model, wrote the update rules, wrote the required scripts, and made the tables and figures. NW and LM carried out the simulations. IG and JK obtained funding for this project.

FUNDING

This work was partially supported by ABACUS, CONACyT grant EDOMEX-2011-C01-165873.

ACKNOWLEDGMENTS

We thank Elisa Domínguez Hüttinger, Marcos Nahmad Bensusan, Víctor Manuel Dávila Borja, Eugenio Azpeitia, Stalin Muñoz, David Rosenblueth, Elena Álvarez Buylla, and Rosa Angélica Castillo Rodríguez for their invaluable advice.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphys.2017.00960/full#supplementary-material

REFERENCES

Azpeitia, E., Muñoz, S., González-Tokman, D., Martínez-Sánchez, M. E., Weinstein, N., Naldi, A., et al. (2017). The combination of the functionalities of feedback circuits is determinant for the attractors number and size in pathway-like boolean networks. *Sci. Rep.* 7:42023. doi: 10.1038/srep42023

Bauer, A. L., Jackson, T. L., Jiang, Y., and Rohlf, T. (2010). Receptor cross-talk in angiogenesis: mapping environmental cues to cell phenotype using a stochastic, Boolean signaling network model. *J. Theor. Biol.* 264, 838–846. doi: 10.1016/j.jtbi.2010.03.025

Bazmara, H., Soltani, M., Sefidgar, M., Bazargan, M., Naeenian, M. M., and Rahmim, A. (2015). The vital role of blood flow-induced proliferation and migration in capillary network formation in a multiscale model of angiogenesis. *PLoS ONE* 10:e0128878. doi: 10.1371/journal.pone.0128878

Bazmara, H., Soltani, M., Sefidgar, M., Bazargan, M., Naeenian, M. M., and Rahmim, A. (2016). Blood flow and endothelial cell phenotype regulation during sprouting angiogenesis. *Med. Biol. Eng. Comput.* 54, 547–558. doi: 10.1007/s11517-015-1341-4

Bentley, K., and Chakravartula, S. (2017). The temporal basis of angiogenesis. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 372:20150522. doi: 10.1098/rstb.2015.0522

Betz, C., Lenard, A., Belting, H.-G., and Affolter, M. (2016). Cell behaviors and dynamics during angiogenesis. *Development* 143, 2249–2260. doi: 10.1242/dev.135616

Biere, A. (2008). Picosat essentials. *J. Satisfiabil. Boolean Model. Comput.* 4, 75–97.

Blancas, A. A., Wong, L. E., Glaser, D. E., and McCloskey, K. E. (2012). Specialized tip/stalk-like and phalanx-like endothelial cells from embryonic stem cells. *Stem Cells Dev.* 22, 1398–1407. doi: 10.1089/scd.2012.0376

Blanco, R., and Gerhardt, H. (2013). VEGF and Notch in tip and stalk cell selection. *Cold Spring Harb. Perspect. Med.* 3:a006569. doi: 10.1101/cshperspect.a006569

Boas, S. E., and Merks, R. M. (2014). Synergy of cell–cell repulsion and vacuolation in a computational model of lumen formation. *J. R. Soc. Interface* 11:20131049. doi: 10.1098/rsif.2013.1049

Bookholt, F., Monsuur, H., Gibbs, S., and Vermolen, F. (2016). Mathematical modelling of angiogenesis using continuous cell-based models. *Biomech. Model. Mechanobiol.* 15, 1577–1600. doi: 10.1007/s10237-016-0784-3

Chappell, J. C., Cluceru, J. G., Nesmith, J. E., Mouillesseaux, K. P., Bradley, V. B., Hartland, C. M., et al. (2016). Flt-1 (vegfr-1) coordinates discrete stages of blood vessel formation. *Cardiovasc. Res.* 111, 84–93. doi: 10.1093/cvr/cvw091

Chen, Q., Jiang, L., Li, C., Hu, D., Bu, J.-W., Cai, D., et al. (2012). Haemodynamics-driven developmental pruning of brain vasculature in zebrafish. *PLoS Biol.* 10:e1001374. doi: 10.1371/journal.pbio.1001374

Chillo, O., Kleinert, E. C., Lautz, T., Lasch, M., Pagel, J.-I., Heun, Y., et al. (2016). Perivascular mast cells govern shear stress-induced arteriogenesis by orchestrating leukocyte function. *Cell Rep.* 16, 2197–2207. doi: 10.1016/j.celrep.2016.07.040

Czirok, A. (2013). Endothelial cell motility, coordination and pattern formation during vasculogenesis. *Wiley Interdiscip. Rev. Syst. Biol. Med.* 5, 587–602. doi: 10.1002/wsbm.1233

Deindl, E., and Schaper, W. (2005). The art of arteriogenesis. *Cell Biochem. Biophys.* 43, 1–15. doi: 10.1385/CBB:43:1:001

del Toro, R., Prahst, C., Mathivet, T., Siegfried, G., Kaminker, J. S., Larrivee, B., et al. (2010). Identification and functional analysis of endothelial tip cell–enriched genes. *Blood* 116, 4025–4033. doi: 10.1182/blood-2010-02-270819

Dubrova, E., and Teslenko, M. (2011). A SAT-based algorithm for finding attractors in synchronous boolean networks. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 8, 1393–1399. doi: 10.1109/TCBB.2010.20

Félix, M.-A., and Barkoulas, M. (2015). Pervasive robustness in biological systems. *Nat. Rev. Genet.* 16, 483–496. doi: 10.1038/nrg3949

Forsythe, J. A., Jiang, B.-H., Iyer, N. V., Agani, F., Leung, S. W., Koos, R. D., et al. (1996). Activation of vascular endothelial growth factor gene transcription by hypoxia-inducible factor 1. *Mol. Cell. Biol.* 16, 4604–4613.

Garg, A., Di Cara, A., Xenarios, I., Mendoza, L., and De Micheli, G. (2008). Synchronous versus asynchronous modeling of gene regulatory networks. *Bioinformatics* 24, 1917–1925. doi: 10.1093/bioinformatics/btn336

Ghaffari, S., Leask, R. L., and Jones, E. A. (2015). Flow dynamics control the location of sprouting and direct elongation during developmental angiogenesis. *Development* 142, 4151–4157. doi: 10.1242/dev.128058

Gianni-Barrera, R., Trani, M., Fontanellaz, C., Heberer, M., Djonov, V., Hlushchuk, R., et al. (2013). VEGF over-expression in skeletal muscle induces angiogenesis by intussusception rather than sprouting. *Angiogenesis* 16, 123–136. doi: 10.1007/s10456-012-9304-y

Gianni-Barrera, R., Trani, M., Reginato, S., and Banfi, A. (2011). To sprout or to split? VEGF, Notch and vascular morphogenesis. *Biochem. Soc. Trans.* 39, 1644–1648. doi: 10.1042/BST20110650

Glaser, D. E., Turner, W. S., Madfis, N., Wong, L., Zamora, J., White, N., et al. (2016). Multifactorial optimizations for directing endothelial fate from stem cells. *PLoS ONE* 11:e0166663. doi: 10.1371/journal.pone.0166663

Glass, D. S., Jin, X., and Riedel-Kruse, I. H. (2016). Signaling delays preclude defects in lateral inhibition patterning. *Phys. Rev. Lett.* 116:128102. doi: 10.1103/PhysRevLett.116.128102

Gödde, R., and Kurz, H. (2001). Structural and biophysical simulation of angiogenesis and vascular remodeling. *Dev. Dyn.* 220, 387–401. doi: 10.1002/dvdy.1118

Heck, T., Vaeyens, M.-M., and Van Oosterwyck, H. (2015). Computational models of sprouting angiogenesis and cell migration: towards multiscale mechanochemical models of angiogenesis. *Math. Model. Nat. Phenom* 10, 108–141. doi: 10.1051/mmnp/201510106

Heil, M., Eitenmüller, I., Schmitz-Rixen, T., and Schaper, W. (2006). Arteriogenesis versus angiogenesis: similarities and differences. *J. Cell. Mol. Med.* 10, 45–55. doi: 10.1111/j.1582-4934.2006.tb00290.x

Ji, J. W., Tsoukias, N. M., Goldman, D., and Popel, A. S. (2006). A computational model of oxygen transport in skeletal muscle for sprouting and splitting modes of angiogenesis. *J. Theor. Biol.* 241, 94–108. doi: 10.1016/j.jtbi.2005.11.019

Jiménez, A., Cotterell, J., Munteanu, A., and Sharpe, J. (2015). Dynamics of gene circuits shapes evolvability. *Proc. Natl. Acad. Sci. U.S.A.* 112, 2103–2108. doi: 10.1073/pnas.1411065112

Kässmeyer, S., Plendl, J., Custodis, P., and Bahramsoltani, M. (2009). New insights in vascular development: vasculogenesis and endothelial progenitor cells. *Anat. Histol. Embryol.* 38, 1–11. doi: 10.1111/j.1439-0264.2008.00894.x

Kerr, B. A., West, X. Z., Kim, Y.-W., Zhao, Y., Tischenko, M., Cull, R. M., et al. (2016). Stability and function of adult vasculature is sustained by Akt/Jagged1 signalling axis in endothelium. *Nat. Commun.* 7:10960. doi: 10.1038/ncomms10960

Kirschner, M., and Gerhart, J. (1998). Evolvability. *Proc. Natl. Acad. Sci. U.S.A.* 95, 8420–8427.

Korn, C., and Augustin, H. G. (2015). Mechanisms of vessel pruning and regression. *Dev. Cell* 34, 5–17. doi: 10.1016/j.devcel.2015.06.004

Kumar, V. S., Binu, S., Soumya, S., Haritha, K., and Sudhakaran, P. (2014). Regulation of vascular endothelial growth factor by metabolic context of the cell. *Glycoconj. J.* 31, 427–434. doi: 10.1007/s10719-014-9547-5

Kurz, H., Burri, P. H., and Djonov, V. G. (2003). Angiogenesis and vascular remodeling by intussusception: from form to function. *Physiology* 18, 65–70. doi: 10.1152/nips.01417.2002

Lloyd-Price, J., Gupta, A., and Ribeiro, A. S. (2012). Robustness and information propagation in attractors of random boolean networks. *PLoS ONE* 7:e42018. doi: 10.1371/journal.pone.0042018

Logsdon, E. A., Finley, S. D., Popel, A. S., and Mac Gabhann, F. (2014). A systems biology view of blood vessel growth and remodelling. *J. Cell. Mol. Med.* 18, 1491–1508. doi: 10.1111/jcmm.12164

Makanya, A. N., Hlushchuk, R., and Djonov, V. G. (2009). Intussusceptive angiogenesis and its role in vascular morphogenesis, patterning, and remodeling. *Angiogenesis* 12, 113–123. doi: 10.1007/s10456-009-9129-5

Müssel, C., Hopfensitz, M., and Kestler, H. A. (2010). Boolnet–an r package for generation, reconstruction and analysis of boolean networks. *Bioinformatics* 26, 1378–1380. doi: 10.1093/bioinformatics/btq124

Naldi, A., Remy, E., Thieffry, D., and Chaouiya, C. (2011). Dynamically consistent reduction of logical regulatory graphs. *Theor. Comput. Sci.* 412, 2207–2218. doi: 10.1016/j.tcs.2010.10.021

Nesmith, J. E., Chappell, J. C., Cluceru, J. G., and Bautch, V. L. (2017). Blood vessel anastomosis is spatially regulated by flt1 during angiogenesis. *Development* 144, 889–896. doi: 10.1242/dev.145672

Norton, K.-A., and Popel, A. S. (2016). Effects of endothelial cell proliferation and migration rates in a computational model of sprouting angiogenesis. *Sci. Rep.* 6:36992. doi: 10.1038/srep36992

Nussenbaum, F., and Herman, I. M. (2010). Tumor angiogenesis: insights and innovations. *J. Oncol.* 2010, 1–24. doi: 10.1155/2010/132641

Palm, M. M., Dallinga, M. G., van Dijk, E., Klaassen, I., Schlingemann, R. O., and Merks, R. M. (2016). Computational screening of tip and stalk cell behavior proposes a role for apelin signaling in sprout progression. *PLoS ONE* 11:e0159478. doi: 10.1371/journal.pone.0159478

Patan, S., Haenni, B., and Burri, P. H. (1996). Implementation of intussusceptive microvascular growth in the chicken chorioallantoic membrane (CAM):: 1. pillar formation by folding of the capillary wall. *Microvasc. Res.* 51, 80–98.

Patan, S., Haenni, B., and Burri, P. H. (1997). Implementation of intussusceptive microvascular growth in the chicken chorioallantoic membrane (CAM): 2. pillar formation by capillary fusion. *Microvasc. Res.* 53, 33–52.

Peirce, S. M. (2008). Computational and mathematical modeling of angiogenesis. *Microcirculation* 15, 739–751. doi: 10.1080/10739680802220331

Qin, D., Trenkwalder, T., Lee, S., Chillo, O., Deindl, E., Kupatt, C., et al. (2013). Early vessel destabilization mediated by Angiopoietin-2 and subsequent vessel maturation via Angiopoietin-1 induce functional neovasculature after ischemia. *PLoS ONE* 8:e61831. doi: 10.1371/journal.pone.0061831

Qutub, A. A., MacGabhann, F., Karagiannis, E. D., Vempati, P., and Popel, A. S. (2009). Multiscale models of angiogenesis. *IEEE Eng. Med. Biol. Mag.* 28, 14–31. doi: 10.1109/MEMB.2009.931791

Qutub, A. A., and Popel, A. S. (2015). "Angiogenesis, computational modeling perspective," in *Encyclopedia of Applied and Computational Mathematics*, ed B. Engquist (Berlin; Heidelberg: Springer Berlin Heidelberg), 58–67.

Saadatpour, A., Albert, I., and Albert, R. (2010). Attractor analysis of asynchronous boolean models of signal transduction networks. *J. Theor. Biol.* 266, 641–656. doi: 10.1016/j.jtbi.2010.07.022

Saadatpour, A., Albert, R., and Reluga, T. C. (2013). A reduction method for Boolean network models proven to conserve attractors. *SIAM J. Appl. Dyn. Syst.* 12, 1997–2011. doi: 10.1137/13090537X

Scharpfenecker, M., Fiedler, U., Reiss, Y., and Augustin, H. G. (2005). The Tie-2 ligand angiopoietin-2 destabilizes quiescent endothelium through an internal autocrine loop mechanism. *J. Cell Sci.* 118, 771–780. doi: 10.1242/jcs.0165

Scianna, M., Bell, C., and Preziosi, L. (2013). A review of mathematical models for the formation of vascular networks. *J. Theor. Biol.* 333, 174–209. doi: 10.1016/j.jtbi.2013.04.037

Siemerink, M. J., Augustin, A. J., and Schlingemann, R. O. (2010). Mechanisms of ocular angiogenesis and its molecular mediators. *Dev. Ophthalmol.* 46, 4–20. doi: 10.1159/000320006

Song, J. W., and Munn, L. L. (2011). Fluid forces control endothelial sprouting. *Proc. Natl. Acad. Sci. U.S.A.* 108, 15342–15347. doi: 10.1073/pnas.1105316108

Vavourakis, V., Wijeratne, P. A., Shipley, R., Loizidou, M., Stylianopoulos, T., and Hawkes, D. J. (2017). A validated multiscale in-silico model for mechano-sensitive tumour angiogenesis and growth. *PLoS Comput. Biol.* 13:e1005259. doi: 10.1371/journal.pcbi.1005259

Veliz-Cuba, A. (2011). Reduction of boolean network models. *J. Theor. Biol.* 289, 167–172. doi: 10.1016/j.jtbi.2011.08.042

Venkatraman, L., Regan, E. R., and Bentley, K. (2016). Time to decide? Dynamical analysis predicts partial tip/stalk patterning states arise during angiogenesis. *PLoS ONE* 11:e0166489. doi: 10.1371/journal.pone.0166489

Vermolen, F., and Javierre, E. (2012). A finite-element model for healing of cutaneous wounds combining contraction, angiogenesis and closure. *J. Mathe. Biol.* 65, 967–996. doi: 10.1007/s00285-011-0487-4

Weinstein, N., Ortiz-Gutiérrez, E., Muñoz, S., Rosenblueth, D. A., Álvarez-Buylla, E. R., and Mendoza, L. (2015). A model of the regulatory network involved in the control of the cell cycle and cell differentiation in the *Caenorhabditis elegans* vulva. *BMC Bioinformatics* 16:1. doi: 10.1186/s12859-015-0498-z

Yadav, L., Puri, N., Rastogi, V., Satpute, P., and Sharma, V. (2015). Tumour angiogenesis and angiogenic inhibitors: A review. *J. Clin. Diagn. Res.* 9, XE01–XE05. doi: 10.7860/JCDR/2015/12016.6135

Zheng, X., Koh, G. Y., and Jackson, T. (2013). A continuous model of angiogenesis: Initiation, extension, and maturation of new blood vessels modulated by vascular endothelial growth factor, angiopoietins, platelet-derived growth factor-b, and pericytes. *Discrete Continuous Dyn. Syst. Ser. B* 18, 1109–1154. doi: 10.3934/dcdsb.2013.18.1109

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

METHODS
published: 24 April 2018
doi: 10.3389/fphys.2018.00431

Automatic Screening for Perturbations in Boolean Networks

Julian D. Schwab[1,2] *and Hans A. Kestler*[1]*

[1] Medical Faculty, Institute of Medical Systems Biology, Ulm University, Ulm, Germany, [2] International Graduate School of Molecular Medicine, Ulm University, Ulm, Germany

A common approach to address biological questions in systems biology is to simulate regulatory mechanisms using dynamic models. Among others, Boolean networks can be used to model the dynamics of regulatory processes in biology. Boolean network models allow simulating the qualitative behavior of the modeled processes. A central objective in the simulation of Boolean networks is the computation of their long-term behavior—so-called attractors. These attractors are of special interest as they can often be linked to biologically relevant behaviors. Changing internal and external conditions can influence the long-term behavior of the Boolean network model. Perturbation of a Boolean network by stripping a component of the system or simulating a surplus of another element can lead to different attractors. Apparently, the number of possible perturbations and combinations of perturbations increases exponentially with the size of the network. Manually screening a set of possible components for combinations that have a desired effect on the long-term behavior can be very time consuming if not impossible. We developed a method to automatically screen for perturbations that lead to a user-specified change in the network's functioning. This method is implemented in the visual simulation framework ViSiBool utilizing satisfiability (SAT) solvers for fast exhaustive attractor search.

Keywords: systems biology, regulatory networks, Boolean networks, dynamic model, simulation, perturbation studies, SAT solving

OPEN ACCESS

Edited by:
Tomáš Helikar,
University of Nebraska-Lincoln,
United States

Reviewed by:
Juilee Thakar,
University of Rochester, United States
Kyle B. Gustafson,
United States Department of the Navy,
United States

***Correspondence:**
Hans A. Kestler
hans.kestler@uni-ulm.de

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: *01 February 2018*
Accepted: *06 April 2018*
Published: *24 April 2018*

Citation:
Schwab JD and Kestler HA (2018) Automatic Screening for Perturbations in Boolean Networks. Front. Physiol. 9:431. doi: 10.3389/fphys.2018.00431

1. INTRODUCTION

Internal and external conditions cause a biological system to change its behavior over time. Mathematical models have become invaluable tools to gain insights into the complex dynamics of biological systems. Boolean networks are one kind of dynamic models based on two-valued logic. Boolean networks can be modeled manually by extraction of Boolean functions from literature resources or inferred automatically from time-series data (Lähdesmäki et al., 2003; Maucher et al., 2011, 2012; Hopfensitz et al., 2012). Simulation of Boolean networks allows for studying various dynamic network properties of the investigated systems. The long-term behavior of the modeled system often corresponds to biologically relevant phenotypes (Naldi et al., 2015). Furthermore, the dynamics of Boolean networks can aid in identifying components that are crucial for these phenotypes. For instance, the effects of depriving or over-representing one element in the system can be measured in the form of changes in the long-term behavior. However, the number of possible perturbations increases rapidly with a larger model size. We developed a method to automatically screen for perturbations that cause a desired effect on the long-term behavior of the system.

There are various tools and frameworks to model, simulate or visualize different types of Boolean networks. The R-package BoolNet comprises a number of simulation algorithms, for instance, attractor search, network perturbation or robustness analysis for synchronous, asynchronous, and probabilistic Boolean networks (Müssel et al., 2010). Additionally, it allows for visualization of dependencies in the network and attractors. However, BoolNet requires programming skills and a basic understanding of the programming language R.

GUI-based software like GinSim (Gonzalez et al., 2006) incorporates different simulation methods for logical models without temporal predicates, including the simulation of manually specified perturbations.

MaBoSS (Stoll et al., 2017) is a tool to simulate Boolean networks stochastically. MaBoSS focuses on a vast number of simulation methods including perturbation studies without the ability to model. We chose to include our methods to automatically screen for perturbations into the existing Java-based framework ViSiBooL (Schwab et al., 2017a). ViSiBooL extends the Boolean network paradigm by temporal predicates and is a light-weight stand-alone modeling and simulation framework. It specifically aims at a straight-forward and easy-to-use modeling and simulation functionality also used by life scientists without any programming skills.

The framework allows to model Boolean networks from scratch and to load existing network models from different sources. Boolean networks can be modeled via graph representations and text-based. The supported SBML-qual standard (Chaouiya et al., 2013) and a simple text network specification format allow for tight interoperability with other common software tools.

In the following we will first briefly define Boolean networks, show how SAT solving (Schöning and Torán, 2013) can be used for attractor search, and then outline our automated screening procedure which can also use temporal predicates in Boolean networks. Finally, we will give some simulation results on a model of the senescence-associated secretory phenotype (SASP).

2. METHODS

2.1. Boolean Networks

Boolean networks are a class of simple logical models that can be used for the modeling of dynamic biological processes such as gene regulation (Kauffman, 1969, 1994). Each component of the modeled system is described by a Boolean variable. It can either be active (true/1) or not (false/0). Dependencies between the different components in the network are described by Boolean functions. The state of a Boolean network with n components at time t is described by a Boolean vector $\mathbf{x}(t) = (x_1(t), \ldots, x_n(t))$. The value of each component x_i at a time t is determined by its corresponding transition function $f_i : \mathbb{B}^n \rightarrow \mathbb{B}$. The successor state $\mathbf{x}(t+1)$ is calculated as follows : $\mathbf{x}(t+1) = (f_1(\mathbf{x}(t), \ldots, f_n(\mathbf{x}(t))$. Here, an exemplary Boolean network with three components x_1, x_2, x_3 and their transition functions is defined : $f_1(\mathbf{x}(t)) = \neg x_1(t)$, $f_2(\mathbf{x}(t) = x_1(t) \vee x_2(t))$, $f_3(\mathbf{x}(t) = x_1(t) \wedge \neg x_2(t))$. There are three major types of Boolean networks -synchronous, asynchronous and probabilistic. In synchronous Boolean networks all variables are updated at the same time. In asynchronous Boolean networks only one randomly chosen variable is updated at each time step $\mathbf{x}(t+1) = (x_1, \ldots, f_i(\mathbf{x}(t)), \ldots, x_n)$, where $i \in [1, n]$ (Harvey and Bossomaier, 1997).

Probabilistic Boolean networks allow for specifying more than one transition function per variable in the network. Each of these functions has a probability of being chosen, where the probabilities of all functions for one variable sum up to 1 (Shmulevich et al., 2002).

The methods presented in the following focus on the simulation of synchronous Boolean networks.

The dynamics of the Boolean networks are studied via examining the transitions from one state to another. The number of states in Boolean networks is finite (2^n in a network with n components). Consequently, the network eventually converges to a recurring number of states after a number of state transitions. These cycles of states are called attractors and represent the long-term behavior of the Boolean network. As already previously mentioned, attractors are of special interest as they often represent biologically relevant behaviors (Naldi et al., 2015). This could be shown in a number of publications successfully using Boolean networks to model the qualitative behavior of a variety of tissues in different organisms (Albert and Othmer, 2003; Fauré et al., 2006; Herrmann et al., 2012; Dahlhaus et al., 2016; Linke et al., 2017; Meyer et al., 2017). All states leading to the same attractor are associated to its so-called basin of attraction (Saadatpour and Albert, 2013). All basins of attraction together comprise the complete number of states.

2.2. Attractor Search and SAT

There are different types of algorithms for attractor search in Boolean networks. Basic algorithms for exhaustive attractor search examine each state. However, these algorithms are demanding in terms of runtime ($\mathcal{O}(2^n)$) and memory ($\mathcal{O}(2^n)$) (Hopfensitz et al., 2013). A number of other algorithms to search for attractors have been proposed. Some of them search efficiently for attractors of length one (Akutsu et al., 2011; Veliz-Cuba et al., 2014). An algorithm that searches for attractors of different length very efficiently is based on SAT-solving (Dubrova and Teslenko, 2011; Naldi et al., 2015). Especially for networks with modest connectivity, this algorithm is more efficient than the exhaustive algorithms that examine every possible state.

Solving a satisfiability (SAT) problem, is basically finding an assignment that satisfies a Boolean formula, i.e., the Boolean formula returns true (Schöning and Torán, 2013). The SAT-solving approach can now be adapted to perturbation studies and the temporal extension in Boolean networks. In the following a basic SAT-based attractor search algorithm is briefly described.

Formally, a state transition can be defined as follows: $T(\mathbf{x}(t), \mathbf{x}(t+1)) = \bigwedge_{i=1}^{n} x_i(t+1) \leftrightarrow f_i(x_1(t), \ldots, x_n(t))$, where n is the number of components in the network. In the algorithm a path—a consecutive sequence of states—is represented by such a Boolean formula. A path of length two in the previously given example network is defined as follows : $T(\mathbf{x}(t), \mathbf{x}(t+1)) = (x_1(t+1) \leftrightarrow \neg x_1(t)) \wedge (x_2(t+1) \leftrightarrow (x_1(t) \vee x_2(t))) \wedge (x_3(t+1) \leftrightarrow x_1(t) \wedge \neg x_2(t))$. A satisfying assignment for this formula corresponds to

a valid, existing path. A SAT-solver can now be used to find all satisfying assignments—each corresponds to one path through the state graph of the Boolean network. Attractors are deduced from these valid paths. Starting with an initial length all valid paths in the Boolean network are determined. First, to compute the the valid solutions for a path the transition formula has to be unfolded. The resulting conjunction of clauses is then solved using a SAT solver.

Next, to detect attractors it is checked whether a state occurs more than once in the path. Obviously, all states between two equal states belong to the attractor. If an attractor is in the path, it is stored and its states are added to the formula as constraints. All other paths including the same attractor are no valid solution anymore. Consequently, the whole basin of attraction of the found attractor is excluded from the search space. If the found path is attractor free, the analyzed sequence of states has to be prolonged to reach the attractor. This procedure is repeated until there is no other valid solution found by the SAT-solver. This means all valid paths to attractors were examined and all existing attractors are found.

In our implementation we used the SAT-solver MINISAT (Eén and Sörensson, 2004) which is based on the idea of conflict-driven backtracking (Marques-Silva and Sakallah, 1999).

2.3. Temporal Predicates in Attractor Search

In synchronous Boolean networks all components are updated at the same time and their value is determined according to the previous state of the network. These assumptions can restrict the modeling or may require hypothetical delay nodes. Biological processes happen on different time scales. In some processes the accumulation of a product over several time steps is required to activate the production of another component. Different components might have different latency periods. The temporal predicates allow the modeling of such latency periods (Schwab et al., 2017a).

In this temporal extension the next state $\mathbf{x}(t+1)$ may not only depend on the previous time step $\mathbf{x}(t)$, but also any other predecessor state $\mathbf{x}(t-\Delta), \Delta = \{1, 2, \ldots, t-1\}$.
For this extension a history of previous values of the relevant components are stored in addition to the current values of the network at time t.

This temporal extension to the synchronous Boolean network model includes two temporal operators. One that allows a direct specification of operations like an accumulation of a gene product over a number of time steps. This operator ALL only evaluates true if a specified term is valid for a given number of time steps. The second operator ANY evaluates true if a term is valid at least once in a specified period of time. The previously described SAT based attractor search is now expanded to include these operators. To find a solution for the unfolded formula of a path each network component at each time step is mapped to another variable. Exemplarily, a path from t to $t+1$ in a network with three components x_1, x_2, x_3 is mapped to six variables $v_1, \ldots, v_6$, where $x_1(t) = v_1, x_2(t) = v_2, \ldots, x_1(t+1) = v_4, \ldots$. Consequently, the formula for the SAT-solver consists of $l \cdot n$ variables, where n is the number of components and l the length of the path. In these temporal Boolean networks the value of a network component does not only depend on the values of the previous state. To enable exhaustive attractor search the mapping had to be changed to reference back to values before the previous time step.
The temporal extension allows the network to stay in a state for more than one time step before moving to another. This prevents searching for multiple occurrences of a state in the path to detect attractors. Not only the states in the path are compared but also their history. True equality of states to detect attractors is only given if their history is also equal.

2.4. Screening for Meaningful Perturbations

Boolean networks can be used for the simulation of various perturbations. Components can be stripped from the system (called knock-down here) or the system can have a surplus of some component (called over-expression here). These behaviors of component x_i can be formally described by

$$x_i(t+1) = \begin{cases} 0 & x_i \text{ is knocked down,} \\ 1 & x_i \text{ is overexpressed,} \\ f_i(\mathbf{x}(t)) & \text{else.} \end{cases}$$

Such interventions of the system may have major effects on its dynamic behavior. The new framework implements various features to investigate the effects of such perturbations.

2.4.1. Single Path Perturbation

Local attractor search from a user-specified initial condition can be modified by knock-down or over-expression of components of interest. The resulting attractor is instantly computed and visualized, which allows for fast comparison of original and perturbation behavior.

2.4.2. Global Network Perturbation

Global effects of perturbations are determined via an extension of the exhaustive search algorithm described in the previous section. Our SAT-solving algorithm was extended to support also fixed components. This implies that in certain cases the Boolean formulae can be simplified. In our procedure this is being performed on a symbolic level prior to conversion into a conjunctive normal form (CNF) for SAT solving.

2.4.3. Automated Screening for Meaningful Perturbations

The two previous methods both rely on user-specified perturbations. However, there are cases in which a user aims at investigating which perturbation shows a wanted effect. For this reason another method was developed.

Here, the user can specify a set of perturbation candidates (**Figure 1C**). Among these candidates, the method searches for all perturbations and combinations of perturbations which show a desired effect.

This effect is also user-defined. Attractors which are intended to exist or not exist under perturbation conditions can be

selected (**Figure 1A,B**). For k user-selected components of interest (**Figure 1C**) all knock-down and/or over-expression combinations of size one up to a user-specified maximum size m are generated. This results in a set P of perturbation combinations to test. Each perturbation $p_i \in P$ is another combination of a number of components in one of the possible perturbation types (knock-down/over-expression). For instance, a set of components $X' = \{x_1, x_2, x_4\}$ is selected and the maximum combination size is two. This results in $P = \{(x_1 = 0), (x_1 = 1), \ldots, (x_1 = 0, x_2 = 0), (x_1 = 0, x_2 =$

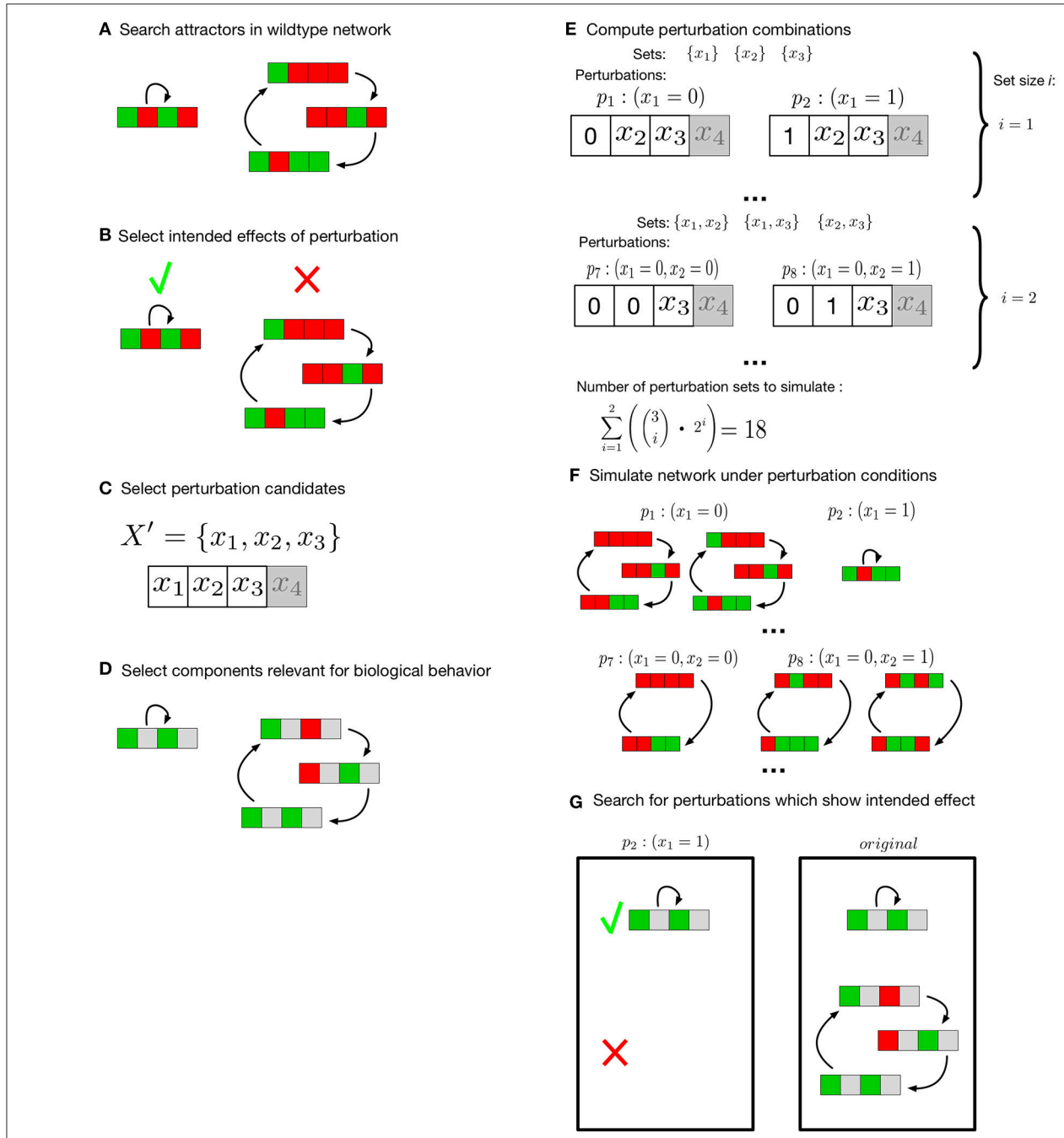


FIGURE 1 | Search for meaningful perturbation effects. Colors green and red represent the Boolean values *true* and *false*, components the user declared irrelevant for the analysis are gray. **(A)** attractors of the unperturbed network are searched exhaustively. **(B)** The user specifies the effects intended by perturbation of the network. **(C)** Components to evaluate under perturbation conditions are selected. **(D)** Selection of components of interest for the attractors under investigation. After the setup by the user **(B–D)** all possible combinations of perturbations are computed **(E)**. Attractors for all perturbation sets are computed **(F)** and compared to the original networks **(G)**. All perturbation sets that match the intended effects are returned.

1), . . .}, $|P| = 18$ (**Figure 1E**) . Next, these $\sum_{i=1}^{m} \binom{k}{i} \cdot 2^i$ combinations of perturbations are evaluated (**Figure 1F**). In this evaluation the longterm behavior of the perturbed network is compared to the longterm behavior of the unperturbed network model (**Figure 1G**). Not all components of the network might be of interest for every description of a biologically relevant behavior. Thus, the user can specify a set of components and the resulting attractors of perturbed network and original network are compared on the basis of these components (**Figure 1D**). Finally, all perturbation combinations pi that match the intended longterm behavior are returned by the algorithm. To increase the simulation speed in our implementation, the different perturbation combinations are evaluated in parallel. The number of parallel instances scales with the number of available cores.

2.5. Biological Example

To illustrate the feasibility of the methods we used the Boolean network described in Meyer et al., 2017, which is a model for the SASP after DNA damage induced senescence. Cellular senescence is a tumor suppressor mechanism which arrests cells before becoming malignant (Coppé et al., 2010; Muñoz-Espín and Serrano, 2014). Senescent cells secrete different factors to attract phagocytic immune cells. Early SASP is probably beneficial to clear the damaged cells. However, once the immune system cannot keep up with the emergence of damaged cells, counteracting the SASP can prevent tissue damage (Meyer et al., 2017). SASP can, for instance, turn senescent fibroblasts into pro-inflammatory cells with the ability to promote tumor progression (Coppé et al., 2010).

The published Boolean network model comprises of two interacting subnetworks—one for DNA damage signaling and one modeling the inflammatory response. The complete model contains 51 components (**Figure 2A**). Attractor search simulation of the network model shows an active immune response after DNA damage (**Figure 2B**). Mayer et al. used the Boolean network model to hypothesize about perturbations that prevent an immune response after DNA damage. These perturbations aim at counteracting the SASP to give the immune system time to catch up. Manual perturbation simulations of the network identified knocking-out NF-κB Essential Modulator (NEMO) is a promising candidate to prevent an immune response—a hypothesis which could be validated by *in-vitro* approaches (Meyer et al., 2017).

3. SIMULATION RESULTS

Evaluation was performed with the previously described Boolean network model of the SASP. In Meyer et al. (2017) different perturbation candidates were manually tested for their deactivation of the major SASP-mediators after DNA damage. Also, attractors had to be analyzed manually to examine feasible candidate perturbations. This approach can be very time consuming for a growing number of candidates to test.

For the evaluation here, we screened the Boolean network model for perturbation candidates that inhibit an immune response after DNA damage. The results were then compared to the results manually investigated by Meyer et al.

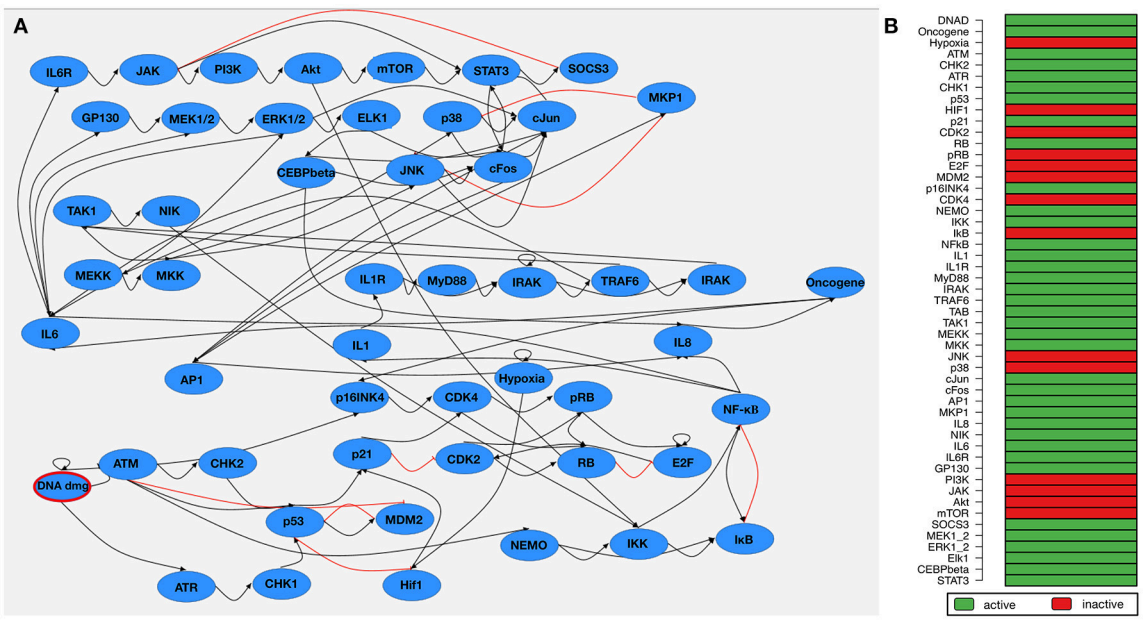


FIGURE 2 | Boolean network of the senescence-associated secretory phenotype (SASP). **(A)** Network wiring of the Boolean network. Blue nodes depict the components of the network model. Black (Red) edges represent a activatory (inhibitory) dependency between the connected components. DNA damage as input node of the network is marked in red. **(B)** Steady-state attractor of the SASP-network under DNA damage conditions (DNA damage input is 1/true). A green (red) row indicates that the corresponding component is active (inactive) in the attractor.

We selected IL-1, IL-6, and IL-8 as components in the Boolean network which are overlapping with the up-regulated factors in the SASP according to Coppé et al., 2010. This correlates with the results of the exhaustive attractor search in the Boolean network under DNA damage conditions (DNA damage input is on, Hypoxia is off, **Figure 2B** shows IL1, IL6, and IL8 are active in the attractors).

For the automatic screening, we selected to remove the attractor (**Figure 3A**) according to their state of the interleukins IL1, IL6, and IL8 (**Figure 3C**). With the perturbation, we aim at blocking the inflammatory response after DNA damage but not at a general inhibition of pro-inflammatory signaling. Thus, we chose each single-component perturbation of all components of the DNA damage signaling subnetwork of the network model as perturbation candidates (19 components which lead to 38 perturbations to test, see **Figure 3B**). During the screening process, attractor search is performed for each candidate perturbation. The attractors are then compared to the original attractors of the network under DNA damage conditions. Perturbations which result in attractors that are differing from the original ones according to there values of a selected set of components (here IL1, IL6, IL8) are returned as valid perturbations.

The screening took 64 s on a MacBook Pro (Intel Core I5, 3.1 GHz and 16GB RAM). The analysis shows a deactivation of the immune response for a knock-out of NEMO, NF-κB, ATM, IKK, or an over-expression of IκB (**Figure 3C**). In addition to the suggested NEMO knock-out of Meyer et al. (2017), the automatic screening reveals four new candidate perturbations - knock-out of ATM, NF-κB, and IKK as well as over-expression of IκB. One possible explanation is their ability to act as SASP-triggering factors, which are mainly relayed through NF-κB. NF-κB has a direct regulatory link to IL1, IL6, and IL8. IKK and IκB both have a direct effect on NF-κB and thus have a regulatory impact on the different Interleukins. NEMO has a regulatory effect on these components via IKK and NF-κB and ATM via NEMO/IKK/NF-κB. The shortest paths from the perturbed components to the Interleukins IL1, IL6, IL8 are between one (perturbation of NF-κB) and four (perturbation of ATM) interactions long. This shows the ability to not only identify direct but also indirect regulators as meaningful perturbation candidates in this complex network by our automatic procedure.

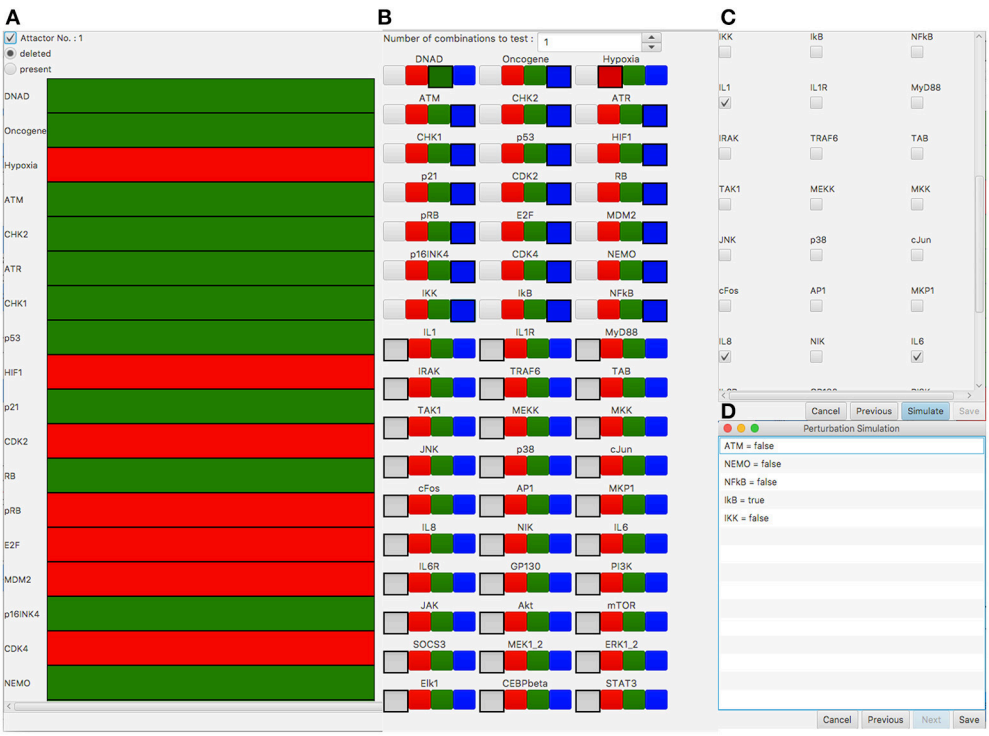


FIGURE 3 | Perturbation screening results in the SASP model (Meyer et al., 2017). **(A)** Attractor under DNA damage conditions which should be removed by perturbation. Green (Red) rows indicate the corresponding component is active (inactive) in the attractor. **(B)** Selection of perturbation candidates to test. Selection of the gray box indicates the component is not of interest for perturbation. Green (Red) means the component is over-expressed (knocked-out). Blue means that both possible perturbations (over-expression/knock-out) are tested for the corresponding component. In this simulation all 19 non-input components that belong to the DNA damage signaling subnetwork of the SASP network are perturbation candidates for knock-out or over-expression. The inputs DNA damage (DNAD) and Hypoxia are fixed to over-expression and knock-out, respectively. The inflammatory signaling part of the network is not selected for perturbation. **(C)** Selection of the genes of interest for attractor comparison. IL1, IL6, and IL8 are selected, which means these components have to be inactive after perturbation. **(D)** Results showing the perturbation candidates that removed the attractor in **(A)** according to the components selected in **(C)**.

4. CONCLUSION

Perturbation studies of Boolean networks can provide more detailed information about the network's inner dynamics. Among others, network perturbation can help to identify therapeutic targets (Saadatpour et al., 2011), to measure a network's capability to compensate mutations (Kwon et al., 2016) or to quantify the robustness of Boolean networks (Schwab et al., 2017b). Furthermore, perturbation of components can be a helpful, assistive tool to check for the expected behavior during the modeling process. Simulation of network perturbation is commonly used in multiple frameworks (Gonzalez et al., 2006; Müssel et al., 2010; Stoll et al., 2017).

The automated screening for perturbations that fulfill user-defined changes in the long-term behavior is—to our best knowledge—a new feature for the analysis of Boolean networks. This feature aims at identifying crucial components for developing a specific long-term behavior. Finding perturbations that eliminate a specified long-term behavior can also be used to screen for therapeutic targets.

These methods were integrated into the Java framework ViSiBooL (Schwab et al., 2017a). ViSiBooL aims at a straightforward and easy-to-use modeling and simulation of Boolean networks. The temporal extension of synchronous Boolean networks allows for a more realistic way of modeling biological processes while maintaining the simple interpretation of synchronous Boolean networks. Moreover, the temporal operators ALL and ANY provide a straight-forward methodology to simplify large terms to model processes over more than one time step. All implemented network perturbation experiments support the temporal network extensions.

AVAILABILITY

The software is available from http://sysbio.uni-ulm.de/?Software:AutoScreenBN

AUTHOR CONTRIBUTIONS

JS and HK: Conceived the software; JS: Implemented the software; JS and HK: Wrote the paper; HK: Supervised the project; HK: Obtained funding for the project.

FUNDING

The research leading to these results has received funding from the European Community Seventh Framework Programme (FP7/2007-2013) under grant agreement nr. 602783, the German Research Foundation (DFG, SFB 1074 project Z1), and the Federal Ministry of Education and Research (BMBF, Gerontosys II, Forschungskern SyStaR, ID 0315894A, and e:Med, SYMBOL-HF, ID 01ZX1407A, CONFIRM, ID 01ZX1708C) all to HK.

REFERENCES

Akutsu, T., Melkman, A. A., Tamura, T., and Yamamoto, M. (2011). Determining a singleton attractor of a Boolean network with nested canalyzing functions. *J. Comput. Biol.* 18, 1275–1290. doi: 10.1109/TCBB.2012.87

Albert, R., and Othmer, H. G. (2003). The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*. *J. Theor. Biol.* 223, 1–18. doi: 10.1016/S0022-5193(03)00035-3

Chaouiya, C., Bérenguier, D., Keating, S. M., Naldi, A., van Iersel, M. P., Rodriguez, N., et al. (2013). SBML qualitative models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools. *BMC Syst. Biol.* 7:135. doi: 10.1186/1752-0509-7-135

Coppé, J.-P., Desprez, P.-Y., Krtolica, A., and Campisi, J. (2010). The senescence-associated secretory phenotype: the dark side of tumor suppression. *Annu. Rev. Pathol. Mech. Dis.* 5, 99–118. doi: 10.1146/annurev-pathol-121808-102144

Dahlhaus, M., Burkovski, A., Hertwig, F., Müssel, C., Volland, R., Fischer, M., et al. (2016). Boolean modeling identifies Greatwall/MASTL as an important regulator in the AURKA network of neuroblastoma. *Cancer Lett.* 371, 79–89. doi: 10.1016/j.canlet.2015.11.025

Dubrova, E., and Teslenko, M. (2011). A SAT-based algorithm for finding attractors in synchronous Boolean networks. *IEEE/ACM Trans. Comput. Biol. Bioinformatics* 8, 1393–1399. doi: 10.1109/TCBB.2010.20

Eén, N., and Sörensson, N. (2004). "An extensible SAT-solver," in *Theory and Applications of Satisfiability Testing*, eds E. Giunchiglia and A. Tacchella (Berlin; Heidelberg: Springer), 502–518.

Fauré, A., Naldi, A., Chaouiya, C., and Thieffry, D. (2006). Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics* 22, e124–e131. doi: 10.1093/bioinformatics/btl210

Gonzalez, A. G., Naldi, A., Sanchez, L., Thieffry, D., and Chaouiya, C. (2006). GINsim: a software suite for the qualitative modelling, simulation and analysis of regulatory networks. *Biosystems* 84, 91–100. doi: 10.1016/j.biosystems.2005.10.003

Harvey, I., and Bossomaier, T. (1997). "Time out of joint: attractors in asynchronous random Boolean Networks," in *Proceedings of the Fourth European Conference on Artificial Life (ECAL97)* (Cambridge, MA: MIT Press), ed C. G. Langton, 67–75.

Herrmann, F., Groß, A., Zhou, D., Kestler, H. A., and Kühl, M. (2012). A Boolean model of the cardiac gene regulatory network determining first and second heart field identity. *PLoS ONE* 7:e46798. doi: 10.1371/journal.pone.0046798

Hopfensitz, M., Müssel, C., Maucher, M., and Kestler, H. A. (2013). Attractors in Boolean networks: a tutorial. *Comput. Stat.* 28, 19–36. doi: 10.1007/s00180-012-0324-2

Hopfensitz, M., Müssel, C., Wawra, C., Maucher, M., Kühl, M., Neumann, H., et al. (2012). Multiscale binarization of gene expression data for reconstructing Boolean networks. *IEEE/ACM Trans. Comput. Biol. Bioinformatics* 9, 487–498. doi: 10.1109/TCBB.2011.62

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.* 22, 437–467. doi: 10.1016/0022-5193(69)90015-0

Kauffman, S. A. (1994). The origins of order. Self-organization and selection in evolution. *J. Evol. Biol.* 7, 518–519.

Kwon, Y.-K., Kim, J., and Cho, K.-H. (2016). Dynamical robustness against multiple mutations in signaling networks. *IEEE/ACM Trans. Comput. Biol. Bioinformatics* 13, 996–1002. doi: 10.1109/TCBB.2015.2495251

Lähdesmäki, H., Shmulevich, I., and Yli-Harja, O. (2003). On learning gene regulatory networks under the Boolean network model. *Mach. Learn.* 52, 147–167. doi: 10.1023/A:1023905711304

Linke, C., Chasapi, A., González-Novo, A., Al Sawad, I., Tognetti, S., Klipp, E., et al. (2017). A Clb/Cdk1-mediated regulation of Fkh2 synchronizes CLB expression in the budding yeast cell cycle. *NPJ Syst. Biol. Appl.* 3:7. doi: 10.1038/s41540-017-0008-1

Marques-Silva, J. P., and Sakallah, K. A. (1999). GRASP: a search algorithm for propositional satisfiability. *IEEE Trans. Comp.* 48, 506–521.

Maucher, M., Kracher, B., Kuhl, M., and Kestler, H. A. (2011). Inferring Boolean network structure via correlation. *Bioinformatics* 27, 1529–1536. doi: 10.1093/bioinformatics/btr166

Maucher, M., Kracht, D. V., Schober, S., Bossert, M., and Kestler, H. A. (2012). Inferring Boolean functions via higher-order correlations. *Comput. Stat.* 29, 97–115. doi: 10.1007/s00180-012-0385-2

Meyer, P., Maity, P., Burkovski, A., Schwab, J., Müssel, C., Singh, K., et al. (2017). A model of the onset of the senescence associated secretory phenotype after DNA damage induced senescence. *PLoS Comput. Biol.* 13:e1005741. doi: 10.1371/journal.pcbi.1005741

Muñoz-Espín, D., and Serrano, M. (2014). Cellular senescence: from physiology to pathology. *Nat. Rev. Mol. Cell Biol.* 15, 482–496. doi: 10.1038/nrm3823

Müssel, C., Hopfensitz, M., and Kestler, H. A. (2010). BoolNet—an R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics* 26, 1378–1380. doi: 10.1093/bioinformatics/btq124

Naldi, A., Monteiro, P. T., Müssel, C., Consortium for Logical Models and Tools, Kestler, H. A., Thieffry, D., et al. (2015). Cooperative development of logical modelling standards and tools with CoLoMoTo. *Bioinformatics* 31, 1154–1159. doi: 10.1093/bioinformatics/btv013

Saadatpour, A., and Albert, R. (2013). Boolean modeling of biological regulatory networks: a methodology tutorial. *Methods* 62, 3–12. doi: 10.1016/j.ymeth.2012.10.012

Saadatpour, A., Wang, R.-S., Liao, A., Liu, X., Loughran, T. P., Albert, I., et al. (2011). Dynamical and structural analysis of a T cell survival network identifies novel candidate therapeutic targets for large granular lymphocyte leukemia. *PLoS Comput. Biol.* 7:e1002267. doi: 10.1371/journal.pcbi.1002267

Schöning, U., and Torán, J. (2013). *The Satisfiability Problem: Algorithms and Analyses, 1st Edn*. Berlin: Lehmanns Media.

Schwab, J., Burkovski, A., Siegle, L., Müssel, C., and Kestler, H. A. (2017a). ViSiBooL-visualization and simulation of Boolean networks with temporal constraints. *Bioinformatics* 33, 601–604. doi: 10.1093/bioinformatics/btw661

Schwab, J., Siegle, L., Kühlwein, S., Kühl, M., and Kestler, H. (2017b). Stability of signaling pathways during aging—a Boolean network approach. *Biology* 6:46. doi: 10.3390/biology6040046

Shmulevich, I., Dougherty, E. R., Kim, S., and Zhang, W. (2002). Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics* 18, 261–274. doi: 10.1093/bioinformatics/18.2.261

Stoll, G., Caron, B., Viara, E., Dugourd, A., Zinovyev, A., Naldi, A., et al. (2017). MaBoSS 2.0: an environment for stochastic Boolean modeling. *Bioinformatics* 33, 2226–2228. doi: 10.1093/bioinformatics/btx123

Veliz-Cuba, A., Aguilar, B., Hinkelmann, F., and Laubenbacher, R. (2014). Steady state analysis of Boolean molecular network models via model reduction and computational algebra. *BMC Bioinformatics* 15:221. doi: 10.1186/1471-2105-15-221

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

ORIGINAL RESEARCH
published: 08 May 2018
doi: 10.3389/fphys.2018.00454

Target Control in Logical Models Using the Domain of Influence of Nodes

Gang Yang[1]*, *Jorge Gómez Tejeda Zañudo*[1,2,3]* and *Réka Albert*[1,4]*

[1] Department of Physics, Pennsylvania State University, University Park, PA, United States, [2] Department of Medical Oncology, Dana-Farber Cancer Institute, Boston, MA, United States, [3] Eli and Edythe L. Broad Institute of MIT and Harvard, Cambridge, MA, United States, [4] Department of Biology, Pennsylvania State University, University Park, PA, United States

Dynamical models of biomolecular networks are successfully used to understand the mechanisms underlying complex diseases and to design therapeutic strategies. Network control and its special case of target control, is a promising avenue toward developing disease therapies. In target control it is assumed that a small subset of nodes is most relevant to the system's state and the goal is to drive the target nodes into their desired states. An example of target control would be driving a cell to commit to apoptosis (programmed cell death). From the experimental perspective, gene knockout, pharmacological inhibition of proteins, and providing sustained external signals are among practical intervention techniques. We identify methodologies to use the stabilizing effect of sustained interventions for target control in Boolean network models of biomolecular networks. Specifically, we define the domain of influence (DOI) of a node (in a certain state) to be the nodes (and their corresponding states) that will be ultimately stabilized by the sustained state of this node regardless of the initial state of the system. We also define the related concept of the logical domain of influence (LDOI) of a node, and develop an algorithm for its identification using an auxiliary network that incorporates the regulatory logic. This way a solution to the target control problem is a set of nodes whose DOI can cover the desired target node states. We perform greedy randomized adaptive search in node state space to find such solutions. We apply our strategy to *in silico* biological network models of real systems to demonstrate its effectiveness.

Keywords: target control, Boolean network, biological network, domain of influence, logical modeling, network dynamics

OPEN ACCESS

Edited by:
Matteo Barberis,
University of Amsterdam, Netherlands

Reviewed by:
Osbaldo Resendis-Antonio,
Universidad Nacional Autónoma de México, Mexico
Maria Suarez-Diez,
Wageningen University & Research, Netherlands
Denis Thieffry,
École Normale Supérieure, France

***Correspondence:**
Gang Yang
yanggangthu@gmail.com
Jorge Gómez Tejeda Zañudo
jgtz@phys.psu.edu
Réka Albert
rza1@psu.edu

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: *04 January 2018*
Accepted: *13 April 2018*
Published: *08 May 2018*

Citation:
Yang G, Gómez Tejeda Zañudo J and Albert R (2018) Target Control in Logical Models Using the Domain of Influence of Nodes. Front. Physiol. 9:454. doi: 10.3389/fphys.2018.00454

1. INTRODUCTION

In cellular systems various molecular species, such as DNA, RNA, proteins and small molecules, interact in diverse ways. The totality of these interactions gives rise to cellular functions. The relationship between molecular interacting systems and cellular functions is studied in the new emerging field of systems biology (Alon, 2006; Palsson, 2006). A promising systems biology methodology is to represent the molecular interacting system as a network, construct a dynamic model of the information propagation on this network, and identify the cellular functions with long-term behaviors of the dynamic model (Palsson, 2006; Newman, 2010; Wang et al., 2012; Barabási and Pósfai, 2016). Various types of dynamical models of biological networks have been built to integrate related experimental results and to reveal the underlying mechanisms of complex

diseases such as cancers, and predict beneficial interventions. Quantitative, mechanistic models, generally using systems of ordinary or partial differential equations, can be highly accurate and provide quantitative information (e.g., response time to a signal, or fold-changes of protein concentrations) (Tyson et al., 2003, 2011; Alon, 2006; Iyengar et al., 2012). These models' widespread use is limited by the scarcity of high-quality quantitative data, such as kinetic and temporal information about individual nodes in the network. Logical models using discrete variables, such as Boolean network models, have the advantage of being scalable and not requiring detailed knowledge of kinetic parameters (Morris et al., 2010; Wynn et al., 2012; Saadatpour and Albert, 2013; Laubenbacher et al., 2014; Abou-Jaoudé et al., 2016; Bloomingdale et al., 2018; Zañudo et al., 2018). An abundance of recent literature has shown that logical models can capture the emergent behaviors of real biological systems, they can generate predictions that are validated by follow-up experiments and they can predict successful intervention strategies (Li et al., 2004; Espinosa-Soto et al., 2004; Mendoza, 2006; Saez-Rodriguez et al., 2007; Naldi et al., 2010; Miskov-Zivanov et al., 2013; Steinway et al., 2015; Albert et al., 2017; Gómez Tejeda Zañudo et al., 2017). For example, logical models of signaling networks that underlie hallmarks of cancer identified the key mechanisms that yield cancer phenotypes and predicted therapeutic interventions that disrupt these phenotypes; many of these predictions were validated experimentally (Grieco et al., 2013; Cohen et al., 2015; Méndez-López et al., 2017; Khan et al., 2017; Kim et al., 2017). Discrete and quantitative models are often consistent in capturing the response repertoire of biological networks (e.g., their potential bistability or response to perturbations) (Kraeutler et al., 2010; Steinway et al., 2016).

Analysis of a logical model entails the determination of the attractors (long-term behaviors) of the system and of the initial states that converge into each attractor (the basins of attraction). Among other uses, this information is used to identify therapeutic interventions as interventions that make a disease attractor unreachable or unstable (Samaga et al., 2010; Abou-Jaoudé et al., 2015; Kim et al., 2017). Attractor identification can be accomplished by simulations of the system's trajectories, determination of all allowed state transitions, or by formal methods such as model checking (Klarner and Siebert, 2015; Abou-Jaoudé et al., 2016), process hitting (Paulevé et al., 2012), or Groebner bases (Laubenbacher et al., 2014). The state space of logical models is finite, but its size scales exponentially with the number of nodes, and thus its full mapping is impossible for systems with many elements. Methods that determine the attractor repertoire of logical models without state space exploration provide a desirable complement to dynamical methods. For example, it was shown that the presence or absence of positive and negative feedback loops in the interaction network puts bounds on the type and number of attractors; e.g., a necessary condition of multistability is the existence of a positive feedback loop (Thomas and D'Ari, 1990; Paulevé and Richard, 2012).

Network control has recently become a popular research topic as it reflects our interest to not only understand an interacting system, but also intervene in it and modify its outcomes (Motter, 2015; Liu and Barabási, 2016). Network control is a broad subject; different underlying models, different control goals and different possible interventions can be considered (Liu and Barabási, 2016). Various control strategies have been designed for both continuous dynamical systems (Liu et al., 2011; Cornelius et al., 2013; Mochizuki et al., 2013; Wells et al., 2015; Wang et al., 2016; Zañudo et al., 2017) and discrete ones (Murrugarra and Dimitrova, 2015; Zañudo and Albert, 2015; Murrugarra et al., 2016; Yang et al., 2016). Of particular interest are the methods that do not require knowledge of the detailed dynamics and parameters of the system, but instead are largely based on the structure of the interaction network and generic assumptions about the functional form of the dependences among variables. In electric circuits modeled by a system of linear ordinary differential equations, it is possible to use graph theoretical methods to identify the set of nodes whose external control can drive the system to any state from any initial condition (Lin, 1974; Liu et al., 2011). For systems with non-linear dynamics, attractor control, that is, to drive the system to one of its natural attractors from any initial condition, has been achieved in several modeling frameworks. Among these, two methods are based on the control of feedback loops: feedback vertex control for ordinary differential equation models (Mochizuki et al., 2013) and stable motif control for logic (Boolean) models (Zañudo and Albert, 2015). However, in biological systems it is not necessary and often not practical to control every component of the system. A more realistic problem is target control, where we assume that the state of the system is characterized by a subset of components and the control goal is to drive these components into desired states. The target control problem has been considered in systems with linear dynamics by Gao et al. (2014), who identified sets of nodes which, if put under suitable (potentially time-varying) external control, drive the target nodes into the desired state.

Despite recent progress in molecular biology, quantitatively manipulating the level of a chemical species is still a challenging problem for experimentalists. Thus any control strategy involving applying time-dependent, variable signals to a target is hard to implement in real systems. However, gene knockout, pharmacological inhibition of proteins and providing sustained external signals have been robustly implemented and demonstrated to be effective intervention strategies (Hopkins and Groom, 2002; Nicholl, 2008; Shalem et al., 2014). Thus we choose our intervention options to be maintaining a sustained state (either absence or abundant activity) in order to make the solution more practical. The effect of such interventions to achieve target control in Boolean network models was previously considered by Klamt et al. (2006) and Samaga et al. (2010). Klamt et al. used the interaction network and regulatory logic to identify the effect of interventions, and determined minimal intervention sets by systematic consideration of all single interventions and combinations of interventions. Samaga et al. made the search for interventions more efficient by using filtering strategies based on the interaction network (e.g., if a candidate intervention source has only negative paths to a target node, then an activating intervention of the source is not useful for activation of the target node) and by grouping equivalent interventions (e.g., if activating a node is sufficient for activating a direct neighbor, then these

interventions are equivalent and only one of them needs to be considered).

Here we propose an alternative and complementary intervention prediction method that uses heuristics based on the system-wide influence of the intervention due both to the connectivity and regulatory logic of the modeled system. Specifically, we base these heuristics on each node's domain of influence (DOI), which identifies which other nodes will adopt a fixed state following an intervention that maintains a sustained state of this node, regardless of the system's initial state. While in general determining the DOI of a node requires exploration of the state space, here we introduce the related concept of logical domain of influence (LDOI) of a node, which can be determined based on the interaction network and the regulatory logic. Specifically, the LDOI is defined on the so-called expanded network introduced in (Albert and Othmer, 2003; Wang and Albert, 2011), which is similar in spirit to a logical interaction hypergraph (Klamt et al., 2006). We use the size and internal consistency of the logic domain of influence (LDOI) to inform a greedy randomized adaptive search to identify the sets of nodes whose DOI can cover the desired target node state (combination).

In the following, we give background information on the Boolean modeling framework and relevant previously-developed concepts such as the expanded network and stable motifs. Then we define the DOI and LDOI of a node or multiple nodes and analyze their properties, such as their internal consistency (or lack thereof) and relationship to dynamic attractors. We then define our target control problem and describe our DOI-based target control strategy using greedy randomized adaptive search in node state space. We finally illustrate the effectiveness of our target control strategy in random ensembles and four *in silico* biological network models.

2. MATERIALS AND METHODS

2.1. Background on Boolean Network Models of Biological Systems

A dynamical model of a biological system starts with the construction of a network (graph) consisting of nodes (also called vertices) that represent the system's elements and edges that specify the pairwise relationships between nodes. In biological networks at the molecular level, nodes are molecular species such as small molecules, RNA, protein, and edges indicate interactions and regulatory relationships. In discrete dynamic (also called logical) models, each node i is characterized by a discrete state variable σ_i, and the vector $(\sigma_1, \cdots, \sigma_n)$ represents the state of the system (Morris et al., 2010; Wynn et al., 2012; Saadatpour and Albert, 2013; Laubenbacher et al., 2014; Abou-Jaoudé et al., 2016; Bloomingdale et al., 2018; Zañudo et al., 2018). The state of the system can be followed in continuous time or at discrete time intervals. In discrete time models, the activity of each node σ_i is described by a regulatory function $\sigma_i(t + \tau_i) = f_i(\sigma_{i_1}(t), \cdots, \sigma_{i_k}(t))$, where $i_1, \cdots, i_k$ are the regulating nodes of i and τ_i is a discrete time delay. The regulatory functions f cannot be constant functions (i.e., cannot yield the same output regardless of the state of the regulators). In models describing signal transduction networks the external signals are represented with source nodes whose regulatory functions depend only on their own state, usually sustaining this state: $\sigma_i(t + \tau_i) = \sigma_i(t)$.

Here we focus on discrete time Boolean network models, where node states are binary, 1(ON) or 0(OFF), and the regulatory function is specified by a truth table or using the Boolean operators AND, OR, NOT (Kauffman, 1969; Glass and Kauffman, 1973). This is motivated by the fact that biological species are frequently observed to demonstrate switch-like behaviors and have highly nonlinear regulations; thus the node state 1 means the molecular species is above a threshold concentration or activity and thus it is able to regulate its targets, and the node state 0 means it is below a threshold concentration or activity and is thus ineffective (Bornholdt, 2008; Wang et al., 2012). Depending on the updating scheme, the time trajectory of the system is simulated deterministically or stochastically. A simple deterministic updating scheme is synchronous updating, where $\tau_i = 1$ for every node (Wang et al., 2012). In this scheme, the system will deterministically evolve from a specific initial state into an attractor, which can be a steady state (fixed point) or a limit cycle, which consists of several states that repeat regularly. Steady states can be interpreted as cell types; limit cycles may correspond to a cell cycle or circadian rhythms. In general asynchronous updating, a commonly used stochastic updating scheme, a random node is selected to be updated at each time step (Glass, 1975). This type of update is motivated by the fact that different biological processes have various timescales, and often the timescales of specific processes are not known (Papin et al., 2005). While limit cycles depend on the specific chosen updating regime, fixed points (steady states) do not depend on the updating scheme (Klemm and Bornholdt, 2005). Stochastic update may lead to attractors that involve irregular repetition of a set of states, called complex attractors.

2.2. The Expanded Network and Its Use in Identifying the Attractor Repertoire of a Boolean Network

The possible combinatorial effect of multiple incoming regulators of a node is important, however, it is not explicitly represented by a regular interaction network. This motivated researchers to develop a concept called the expanded network, which integrates the original network with the regulatory rules of each node (Albert and Othmer, 2003; Wang and Albert, 2011). We illustrate the expanded network with the example in **Figure 1** , which consists of five nodes, node 0, 1, 2, 3, and 4 with the regulatory functions f_0 = NOT σ_3, f_1 = (NOT σ_0) OR σ_3, f_2 = NOT σ_1, f_3 = (NOT σ_2) OR (NOT σ_4), f_4 = σ_0 OR σ_1. First, we denote each original node i by n_i in the expanded network, and we introduce a complementary node for each original node in the system to represent the negation (deactivation) of the original node, denoted by $\sim n_i$ (Wang and Albert, 2011). As the NOT function is a unary operator, all the NOT functions are replaced by the negated state of the respective node (i.e., its complementary node) in each Boolean regulatory function. Edges are introduced in the expanded network to

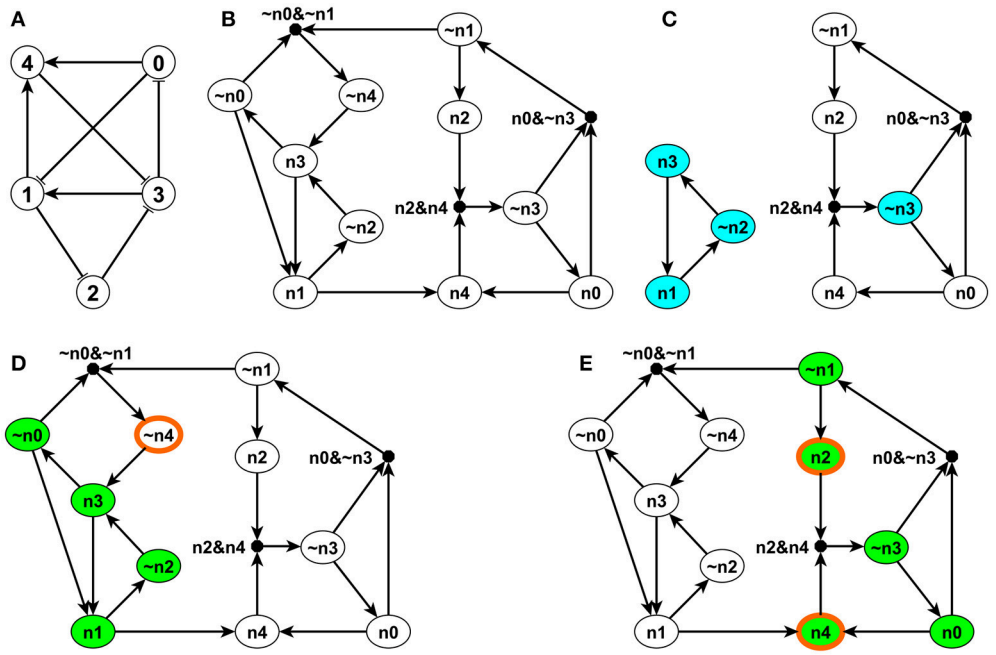


FIGURE 1 | Illustration of the expanded network, stable motifs and logic domain of influence on a simple example. The network is shown in panel **(A)**. Each edge with an arrow represents activation and each edge with a flat bar represents inhibition. The Boolean regulatory functions are specified as follows: f_0 = NOT σ_3, f_1 = (NOT σ_0) OR σ_3, f_2 = NOT σ_1, f_3 = (NOT σ_2) OR (NOT σ_4), $f_4 = \sigma_0$ OR σ_1. Panel **(B)** shows the expanded network of this example. Each node i in panel **(A)** has a correspondent n_i and its complementary node $\sim n_i$ in panel **(B)**. (Note that n_i is labeled as *ni* in panel **(B)** to be more visible). A composite node is drawn as a filled black circle and & represents the AND logic operator. Panel **(C)** indicates the stable motifs; each blue node is a single-node core of the corresponding stable motif. Panels **(D,E)** show the LDOI of $\{\sim n_4\}$ and $\{n_2, n_4\}$, respectively, overlaid over the expanded network. Nodes with thick orange boundary are the sustained interventions and the green nodes are their LDOI.

represent the thus-transformed regulatory functions so that every edge represents a positive regulatory relationship in the expanded network. For example, f_0 = NOT σ_3 implies the rule for the original node n_0 as f_{n_0} = NOT n_3 = $\sim n_3$, and thus a corresponding edge is drawn from $\sim n_3$ to n_0 in the expanded network. The Boolean regulatory function for the complementary (negated) node is the logical negation of the regulatory function of the original node. In this example, $f_{\sim n_0}$ = NOT (NOT n_3) = n_3 and thus a corresponding edge is drawn from n_3 to $\sim n_0$ in the expanded network.

Second, to differentiate OR rules from AND rules when multiple edges point toward the same target node, we introduce a composite node for each set of edges that are linked by an AND function (Wang and Albert, 2011). In order to uniquely determine the edges of the expanded network, the regulatory functions need to be specified in disjunctive normal form, that is, a disjunction of conjunctive clauses (in other words, grouped AND clauses separated by OR clauses). For example, (*A* AND *B*) OR (*A* AND *C*) is in a disjunctive normal form, while its equivalent form *A* AND (*B* OR *C*) is not. The desired disjunctive normal form can be formed by a disjunction of all conditions that give output 1 in the Boolean table and then simplified to the disjunction of prime implicants (Blake canonical form) by the Quine-McCluskey algorithm (McCluskey, 1956). Now we add a composite node for each AND clause in the Boolean regulatory function, denoted by a filled black circle in **Figure 1B**. We add edges from the non-composite nodes of the expanded network that form this clause to this composite node. For example, the composite node $\sim n_0$&$\sim n_1$ in the left upper part of **Figure 1B** represents the expression (NOT n_0) AND (NOT n_1). The expanded network has edges from $\sim n_0$ to the composite node and from $\sim n_1$ to the composite node. This composite node expresses the regulatory function of the complementary node $\sim n_4$, namely, $f_{\sim n_4}$ = NOT f_{n_4} = NOT n_0 AND NOT n_1 = $\sim n_0$ AND $\sim n_1$. To reflect this, the expanded network contains an edge from this composite node to $\sim n_4$. Now the benefit of introducing complementary and composite nodes is evident: one can read all the regulatory functions from the topology of the expanded network. The NOT rule is indicated by a complementary node, the AND rule is indicated by a composite node with multiple regulators, while all the other edges represent independent activation (parts of an OR function). Moreover, the expanded network also incorporates the negations of the regulatory functions. Thus, for each node *i*, the expanded network reflects the condition that needs to be satisfied in order for $\sigma_i = 1$ (in the incoming edges of the node n_i) and the condition that needs to be satisfied in order for $\sigma_i = 0$ (in the incoming edges of the node $\sim n_i$).

As the expanded network encapsulates the regulatory logic that determines the network dynamics, it can serve as a basis

for attractor analysis. One approach is through analyzing the stable motifs of the expanded network (Zañudo and Albert, 2013). A stable motif is defined as the smallest strongly connected component (SCC) satisfying the following two properties: (1) The SCC cannot contain both a node and its complementary node and (2) If the SCC contains a composite node, it must also contain all of its input nodes (Zañudo and Albert, 2013). The first requirement guarantees that the SCC does not contain any conflict in node states and the second requirement guarantees that all the conditional dependence is satisfied and the SCC is self-sufficient in maintaining each node state inside the stable motif. Thus the stable motif represents a group of nodes that can sustain their states irrespective of outside nodes' states. The corresponding node states implied by the stable motif can be directly read out: an original node represents the ON (1) state and a complementary node represents the OFF (0) state (Zañudo and Albert, 2013). For example, in the left part of **Figure 1C**, node $n_1, \sim n_2$, and n_3 form a stable motif, representing that node 1 and node 3 are ON and node 2 is OFF. There is a strong correspondence between stable motifs and the attractors of the system. Specifically, there is a one-to-one correspondence between a sequence of stable motifs and a fixed point or a partial fixed point (a part of a complex attractor). A partial fixed point is defined as a true subset of all the nodes whose respective state remains unchanged after being updated regardless of the states of the nodes excluded from this subset (Zañudo and Albert, 2013).

2.3. The Domain of Influence of a Sustained Node State

We define the DOI of an intervention that maintains a sustained node state as all the node states that will be stabilized (i.e., attain a stationary value) in the long term under the influence of this intervention for all initial conditions in any updating regime. Mathematically, $\mathcal{D}(\sigma_i = \tilde{\sigma}_i) = \{\sigma_j = \tilde{\sigma}_j : \sigma_j(t) = \tilde{\sigma}_j \text{ as } t \to \infty \text{ for any } (\sigma_1(t=0), ..., \sigma_k(t=0)) \text{ when } \sigma_i(t) = \tilde{\sigma}_i \text{ for any } t > 0\}$, where $\sigma_i(t) = \tilde{\sigma}_i$ is the intervention, $\tilde{\sigma}_i = 0$ represents knockout or suppression and $\tilde{\sigma}_i = 1$ represents sustained activation, $\tilde{\sigma}_j$ represents a node state fixed by the intervention, and $(\sigma_1(t=0), ..., \sigma_k(t=0))$ represents the initial condition of all the nodes of the system. We do not include the intervention node state $\sigma_i = \tilde{\sigma}_i$ in its own DOI, unless the node is sufficient to maintain the corresponding node state in the long term even in the absence of a sustained intervention. Notice that there is one-to-one correspondence between a node state $\sigma_i = \tilde{\sigma}_i$ and a non-composite node n^{ex} in the expanded network : $\sigma_i = 1$ corresponds to a normal node n_i in the expanded network and $\sigma_i = 0$ corresponds to a negation node $\sim n_i$. Thus we use the two notations interchangeably, that is, $\sigma_j = 1 \in \mathcal{D}(\sigma_i = 1)$ is equivalent to $n_j \in \mathcal{D}(n_i)$ and $\sigma_j = 0 \in \mathcal{D}(\sigma_i = 0)$ is equivalent to $\sim n_j \in \mathcal{D}(\sim n_i)$.

The DOI of a node is difficult to calculate because it entails determining the common part of all attractors of a dynamical system to identify the nodes whose states stabilize due to the considered intervention. As an alternative to this computationally hard problem, we define a related concept called the LDOI of an intervention that maintains a sustained node state. The LDOI consists of all the node states that, for any initial condition, are stabilized by the first update of the corresponding node in an updating regime that preserves the level order (breadth first search order) of the expanded network. An updating regime preserves the level order if all the nodes in the nth level are updated at least once before updating any node in the $(n+1)$th level (see details in Supplementary Material 2.1). We denote the LDOI of a node state σ_i as $\mathcal{LD}(\sigma_i = \tilde{\sigma}_i)$. We define the LDOI of an empty set to be an empty set, $\mathcal{LD}(\emptyset) = \emptyset$. This is consistent with the definition as an updating order preserving the level order starting from a null set can start from any node, and a node will not achieve a stationary state upon its very first update for all initial conditions unless its regulatory function is a constant. Source nodes remain in their initial state, which nevertheless will be different for different initial conditions.

2.4. Determining the Logical Domain of Influence of a Sustained Node State

We propose to find the LDOI of a node state by doing a modified breadth first search (BFS) on the expanded network (see the pseudocode in Supplementary Material section 1.1). In order to find the LDOI of $\sigma_i = \tilde{\sigma}_i$, we start the search from the corresponding node n_i on the expanded network if $\sigma_i = 1$ or we start the search from the complementary node $\sim n_i$ if $\sigma_i = 0$. If we meet another non-composite node, we add this node to the LDOI; if we meet a composite node, we add this composite node only if all of its parent nodes (i.e., regulators) are already part of the LDOI. This is due to the fact that any edge from a node to a non-composite node represents a sufficient relationship and any edge from a node to a composite node represents a necessary relationship. We keep searching on the expanded network until no new nodes can be added to the LDOI. For example, in **Figure 1B**, one can readily see that $\mathcal{LD}(\sigma_1 = 1) \equiv \mathcal{LD}(n_1) = \{n_4, \sim n_2, n_3, n_1, \sim n_0\}$ following the described search procedure. The first difference from a normal BFS to find a connected component starting from a node is that we put an extra rule for including a composite node. Another subtle difference is that we do not include the starting point unless we visit this starting point again in our search process.

During the search process, there is a possibility that we meet the negation of the starting point. This reflects the possibility that a node state can indirectly lead to the opposite state through a negative feedback loop. This outcome represents a conflict with the original intervention. We do not add this node to the LDOI because we assume that the intervention can sustain the original node state, thus the opposite state is not reachable. This truncation of the LDOI to avoid including the negation of the starting node state ensures that the LDOI will not contain a node which is the negation of an already visited node. Mathematically, if a non-composite node $n_i^{ex} \in \mathcal{LD}(n_j^{ex})$, then n_j^{ex} is sufficient to activate n_i^{ex}, i.e., the long-term logical rule for n_i^{ex} can be expressed in the form $n_i^{ex} = n_j^{ex}$ OR $\cdots$; this implies $\sim n_i^{ex} = \sim n_j^{ex}$ AND $\cdots$, i.e., $\sim n_j^{ex}$ is necessary to activate $\sim n_i^{ex}$. Thus any conflict between n_i^{ex} and $\sim n_i^{ex}$ will occur after the conflict between n_j^{ex} and $\sim n_j^{ex}$ during the search process. This

truncation of the LDOI is the third difference compared with a normal BFS.

For example, in the network of **Figure 1D**, the LDOI of the complementary node $\sim n_4$ includes nodes $n_3, \sim n_0, n_1, \sim n_2$ following the search procedure. From n_1 one can also reach node n_4, which is the negation of the considered intervention. Thus we stopped this branch of searching based on our truncation rule. Since there are no more nodes that can be added, we conclude that $\mathcal{LD}(\sim n_4) = \{n_3, \sim n_0, n_1, \sim n_2\}$.

Our LDOI search procedure is equivalent to doing a simulation on the expanded network. If we update the system corresponding to the BFS order of the expanded network starting from the intervention node (i.e., we update node i if we visited n_i^{ex} on the expanded network), all the updated nodes are guaranteed to stabilize in the corresponding visited state on the expanded network, i.e., as in the LDOI of that node. In the example of **Figure 1**, as discussed above, $\mathcal{LD}(n_1) = \{n_4, \sim n_2, n_3, n_1, \sim n_0\}$. If we update the nodes in the order 4, 2, 3, 1, 0, each node will stabilize in the state as in $\mathcal{LD}(n_1)$. We note that this does not put a restriction on the updating regime: if we update the system in an arbitrary order, each node in the LDOI of the given sustained intervention will attain a stationary state in the first update after all of its regulators included in the LDOI have been updated once. For example, if we fixed the node 1 to be ON and we perform rounds of update of the nodes in the order 0, 1, 2, 3, 4, nodes 2, 3, and 4 will be stabilized in the first round of updating, while nodes 0 and 1 will be stabilized in the second round.

The difference between the LDOI and DOI is that LDOI requires the nodes to be stabilized when being updated for the first time, while DOI just requires the nodes to be stabilized in finite time. Thus one can see that the LDOI of a node will be a subset of the DOI of a node. In many cases the two concepts give the same result. Two exceptions are illustrated in **Figure 2**. In both cases the DOI of an intervention contains more nodes than the LDOI of this intervention. This is because certain nodes may stabilize not because of the influence of the intervention but because of the collective effect of two inconsistent feedback loops or because of a stable motif stabilized by an oscillation. In the network of **Figure 2A**, the three regulators of node B are independent and the network includes both a positive and a negative feedback loop. To analyze the LDOI of $A = 1$, taking the feedback effect of C and D on B into consideration, the regulatory function of B is simplified into $\sigma_B(t + \tau_B) = \sigma_B(t - \tau_C)$ OR NOT $(\sigma_B(t - \tau_D))$, where τ_i is the discrete time delay for node i, as introduced in section 2.1. This regulatory function admits a constant solution $\sigma_B = 1$ regardless of the values of the time delays (Saadatpour et al., 2010; Azuma et al., 2014). It may additionally admit an oscillatory solution for strict relationships among the time delays. In the cases where there is no oscillatory solution, for example in the cases where only one node can change state at a time, $\mathcal{D}(A) = \{B, C, \sim D\}$, as the stabilization of B leads to the stabilization of C and D as well. However, $\mathcal{LD}(A) = \emptyset$ as the activation of the composite node requires nodes $A, \sim C, \sim D$ on the expanded network shown in **Figure 2B** and thus we cannot add the composite node to the LDOI of node A. In the example shown in **Figure 2C**, the two regulators are independent for node B, $\mathcal{D}(C) = \{B\}$ as the negative feedback loop of node A will make A oscillate, but B will stabilize into the ON state after the first time that A visits the ON state and activates B, while $\mathcal{LD}(C) = \emptyset$ for the same reason as in the last example.

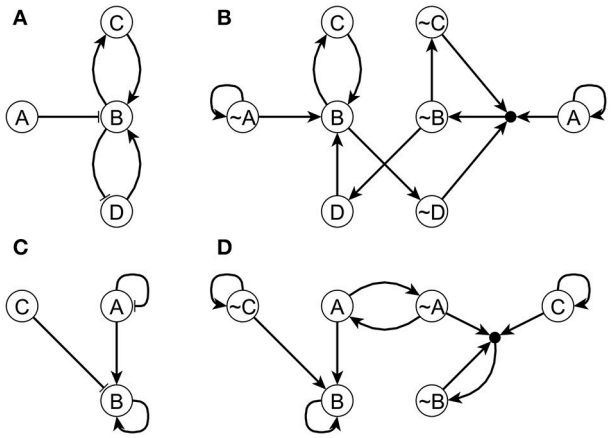


FIGURE 2 | Two example networks **(A,C)** and their respective expanded networks **(B,D)** that illustrate the difference between DOI and LDOI. In both networks, an edge with an arrowhead represents activation while an edge with flat bar represents inhibition. Implicit positive self-loops stemming from the assumed sustained states of source nodes are not shown in panels **(A,C)**. In panel **(A)** the regulatory functions are $f_A = A$, $f_B =$ (NOT A) OR C OR D, $f_C = B$, $f_D =$ NOT B . When A = 1 the system has a fixed point $\sigma_B = \sigma_C = 1, \sigma_D = 0$. In panel **(C)** the regulatory functions are $f_A =$ NOT A, $f_B = A$ OR B OR NOT C, $f_C = C$. When $\sigma_C = 1$ the system has a complex attractor in which A oscillates and $\sigma_B = 1$.

2.5. Properties of the Logical Domain of Influence of a Sustained Node State

In order to further illustrate the concept of LDOI, we discuss a few of its properties and its relationship with established concepts in Boolean dynamics. The LDOI of a node state is mathematically equivalent to the three-valued logical steady state that results when this node state is fixed (Klamt et al., 2006; Samaga et al., 2010). Here the three values are 0, 1, and unknown (for nodes who do not attain a stationary state solely due to the original node's fixed state). The LDOI of a node state is also equivalent to the set of nodes, and corresponding states, that are identified using network reduction techniques [i.e., by iteratively substituting the fixed node state(s) into the regulatory function(s) of target nodes] (Bilke and Sjunnesson, 2001; Naldi et al., 2012; Saadatpour et al., 2013). Previous analysis (Samaga et al., 2010; Saadatpour et al., 2013) identified that if node i has a single outgoing edge, and is a sufficient regulator of its sole target node, j, the LDOI of the ON state of i contains the LDOI of the ON state of j. Here we study in general the possible inclusion relationship between the logic domains of influence of two node states $\sigma_i = \tilde{\sigma}_i$ and $\sigma_j = \tilde{\sigma}_j$ in the case when $\sigma_j = \tilde{\sigma}_j \in \mathcal{LD}(\sigma_i = \tilde{\sigma}_i)$ or $n_j^{ex} \in \mathcal{LD}(n_i^{ex})$ in the expanded network notation, where n_i^{ex} and n_j^{ex} represent any non-composite node in the expanded network. In a directed graph, if node n_j is a reachable from node n_i, all descendants of n_j will also be reachable from n_i; indeed one can

easily prove this by contradiction. However, due to the special properties of the expanded network and the truncation of the LDOI, this inclusion relationship $\mathcal{LD}(n_j^{ex}) \subseteq \mathcal{LD}(n_i^{ex})$ is not generally true for the expanded network. It is possible that $n_j^{ex} \in \mathcal{LD}(n_i^{ex})$, however, $\sim n_i^{ex} \in \mathcal{LD}(n_j^{ex})$. In this case, by definition of the LDOI, we won't allow the negation of a node state to be part of the LDOI of a node state. For example, $n_1 \in \mathcal{LD}(\sim n_4)$, however, $n_4 \in \mathcal{LD}(n_1)$. Thus $\mathcal{LD}(n_1) \not\subset \mathcal{LD}(\sim n_4)$.

If we add an additional restriction on the two nodes, this inclusion relationship will hold the same way as for descendants in a directed graph. To be specific, the *first key property of the LDOI* is, if the node state $\sigma_i = \tilde{\sigma}_i$ and $\sigma_j = \tilde{\sigma}_j$, corresponding to the two non-composite node n_i^{ex} and n_j^{ex} on the expanded network, are both included in the same (partial) fixed point and $n_j^{ex} \in \mathcal{LD}(n_i^{ex})$, the LDOI of n_j^{ex} will be a subset of the LDOI of n_i^{ex}, i.e., $\mathcal{LD}(n_j^{ex}) \subseteq \mathcal{LD}(n_i^{ex})$. (Recall that a partial fixed point is a subset of nodes whose respective state remains unchanged after being updated regardless of the states of the nodes excluded from this subset.) The reason why the inclusion relationship holds is that node states in a (partial) fixed point stabilize in the long term, thus they will not lead to a situation with opposing behavior $n_j^{ex} \in \mathcal{LD}(n_i^{ex})$ and $\sim n_i^{ex} \in \mathcal{LD}(n_j^{ex})$. This restriction can be weakened to only require that node state n_i^{ex} is in a (partial) fixed point. The reason is that if $n_j^{ex} \in \mathcal{LD}(n_i^{ex})$ and n_i^{ex} is in a (partial) fixed point, then n_j^{ex} must also be in the same (partial) fixed point, or be a node whose state stabilizes due to the nodes in the partial fixed point. Also, as one or more stable motifs are part of a (partial) fixed point, the conclusion will be true if one replaces "(partial) fixed point" by "stable motif" in the above statement. In the example of **Figure 1**, as nodes n_1, $\sim n_2$ and n_3 form a stable motif and its corresponding (partial) fixed point is $(\sigma_1, \sigma_2, \sigma_3) = (1, 0, 1)$ as shown in **Figure 1C**, which also lead to the stabilization of the remaining two nodes as $\sigma_0 = 0$ and $\sigma_4 = 1$, thus $n_3 \in \mathcal{LD}(n_1)$ implies that $\mathcal{LD}(n_3) \subseteq \mathcal{LD}(n_1)$. In fact, $\mathcal{LD}(n_3) = \mathcal{LD}(n_1) = \{n_4, \sim n_2, n_3, n_1, \sim n_0\}$. Also $n_4 \in \mathcal{LD}(n_1)$ implies that $\mathcal{LD}(n_4) \subseteq \mathcal{LD}(n_1)$. Note that only n_1 is part of the stable motif or partial fixed point in the latter example, n_4 is not.

As stable motifs represent generalized positive feedback loops of the Boolean network (Zañudo and Albert, 2013), we explore the relationship between stable motifs and the LDOI of a node state. The *second key property of LDOI* is, if the LDOI of a node state contains this node state itself, the LDOI contains a stable motif. As the LDOI of a node state only contains the node state itself if we meet this node during the search process on the expanded network, this indicates the existence of a positive feedback loop, which is the intuition why this proposition holds. (A sketch of proof from the dynamical standpoint is included in Supplementary Material section 2.2). For example, $n_1 \in \mathcal{LD}(n_1)$ implies that there exists a stable motif contained in $\mathcal{LD}(n_1)$, indeed, $SM_1 = \{n_1, \sim n_2, n_3\} \subseteq \mathcal{LD}(n_1)$.

2.6. The Domain of Influence of a Node State Set

Now we generalize the concept of DOI of a single node state to DOI of a node state set (i.e., a set of nodes, each in a sustained state). We define the DOI of a node state set as all the node states that can be stabilized in the long term by the given set of node states under all initial conditions in any updating regime. Mathematically, $\mathcal{D}(\{\sigma_i = \tilde{\sigma}_i\}) = \{\{\sigma_j = \tilde{\sigma}_j\} : \sigma_j(t) = \tilde{\sigma}_j \text{ as } t \to \infty \text{ for any } (\sigma_1(t=0), ..., \sigma_k(t=0)) \text{ when } \sigma_i(t) = \tilde{\sigma}_i \text{ for any } t > 0\}$, where $\{\sigma_i(t) = \tilde{\sigma}_i\}$ represents the intervention consisting of a specific set of node states. Note that the following two notations are equivalent: $\mathcal{D}(\{\sigma_i = \tilde{\sigma}_i\}) \equiv \mathcal{D}(\{n_i^{ex}\})$. Similarly, we define the LDOI of a node state set, $\mathcal{LD}(\{\sigma_i = \tilde{\sigma}_i\})$, as all the nodes that can be stabilized by the first update in any BFS order-preserving (on the expanded network) update order starting from this given set of node states under all initial conditions. As in the single node state case, the LDOI of a node state set will be a subset of the DOI of the same node state set.

The LDOI of a node state set can be determined by a modified BFS on the expanded network, now using multiple starting points. This does not add complexity to the iterative implementation of BFS: we just need to initialize the queue with the set of given node states. Similar to the case of finding the LDOI of a single node state, we need to deal with the conflicts that may occur during the search process. To be precise, conflict means that during the search we visit a node state that is the negation of a node state included in the intervention. Two types of conflict can arise. First, a node state in the given set may be impacted by negative feedback and have a LDOI that was truncated to avoid containing its own negation. Second, the LDOI of two node states n_i^{ex} and n_j^{ex} may have the property $\sim n_i^{ex} \in \mathcal{LD}(n_j^{ex})$ or $\sim n_j^{ex} \in \mathcal{LD}(n_i^{ex})$, or both. In other words, node i may regulate node j (or *vice versa*) in a way that is incompatible with the intervention (e.g., a node whose sustained activity is part of the intervention may negatively regulate another node whose sustained activity is part of the intervention). We call intervention sets that have either type of conflict incompatible sets; we refer to the rest of the intervention sets as compatible sets. Similarly to the truncation we did to find the LDOI of a single node state, we do not include any node state that is the negation of any node state given in the intervention set and we stop searching that branch. We note that this truncation strategy avoids any following conflict. For example, if $n_C \in \mathcal{LD}(n_A)$ and $\sim n_C \in \mathcal{LD}(n_B)$, then one may expect that the LDOI of the set $\{n_A, n_B\}$ will have a conflict between n_C and $\sim n_C$. However, $n_C \in \mathcal{LD}(n_A)$ implies that $\sim n_C$ requires $\sim n_A$, this means that meeting the conflict between n_C and $\sim n_C$, must be after meeting the conflict between n_A and $\sim n_A$, which is avoided by our truncation strategy.

For a compatible set $\{n_i^{ex}\} \equiv \cup_i n_i^{ex}$, it is guaranteed that $\cup_i \mathcal{LD}(n_i^{ex}) \subseteq \mathcal{LD}(\cup_i n_i^{ex})$. For example, as shown in **Figure 1E**, the node set $\{n_2, n_4\}$ is a compatible node set as $\mathcal{LD}(n_2) = \emptyset$, $\mathcal{LD}(n_4) = \emptyset$ and $\mathcal{LD}(\{n_2, n_4\}) = \{\sim n_3, n_0, n_4, \sim n_1, n_2\}$. Note $\mathcal{LD}(n_2) \cup \mathcal{LD}(n_4) \subseteq \mathcal{LD}(\{n_2, n_4\})$. However, for an incompatible set, we just know that the situation $\cup_i \mathcal{LD}(n_i^{ex}) \subsetneq \mathcal{LD}(\cup_i n_i^{ex})$ cannot happen and all the remaining situations are possible. In the network of **Figure 1**, node set $\{n_2, \sim n_4\}$ is an incompatible node set as $\mathcal{LD}(n_2) = \emptyset$, $\mathcal{LD}(\sim n_4) = \{n_3, \sim n_0, n_1, \sim n_2\}$, and $\mathcal{LD}(\{n_2, \sim n_4\}) = \{n_3, \sim n_0, n_1\}$. Note that neither n_4 nor $\sim n_2$ are included in $\mathcal{LD}(\{n_2, \sim n_4\})$ due to the truncation rule and $\mathcal{LD}(\{n_2, \sim n_4\}) \subsetneq \mathcal{LD}(n_2) \cup \mathcal{LD}(\sim n_4)$. Node set $\{\sim n_1, n_3\}$ is another incompatible set as $\mathcal{LD}(\sim n_1) = \{n_2\}$, $\mathcal{LD}(n_3) =$

$\{\sim n_0, n_1, \sim n_2, n_4, n_3\}$ and $\mathcal{LD}(\{\sim n_1, n_3\}) = \{n_2, \sim n_0, \sim n_4, n_3\}$. Note that $\mathcal{LD}(\{\sim n_1, n_3\}) \not\subset \mathcal{LD}(\sim n_1) \cup \mathcal{LD}(n_3)$, and $\mathcal{LD}(\sim n_1) \cup \mathcal{LD}(n_3) \not\subset \mathcal{LD}(\{\sim n_1, n_3\})$.

The properties of the LDOI of a single node can also be generalized to the LDOI of a given node set. For the first key property, let $S_j = \{\sigma_j = \tilde{\sigma}_j\}$ and $S_i = \{\sigma_i = \tilde{\sigma}_i\}$ be two sets of node states, if S_i is a subset of any (partial) fixed point and $S_j \subseteq \mathcal{LD}(S_i)$, then $\mathcal{LD}(S_j) \subseteq \mathcal{LD}(S_i)$. The intuition is similar, the requirement restricting us to consider those nodes which can be stabilized in the long term, that is, we rule out the possibility of S_i being an incompatible node set. For example in **Figure 1** consider $S_i = \{\sim n_3\}$ and $S_j = \{n_2, n_4\}$. As $\sim n_3$ is part of the stable motif $SM_2 = \{n_0, \sim n_1, n_2, \sim n_3, n_4\}$, corresponding to the fixed point $(\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4) = (1, 0, 1, 0, 1)$, $S_j \subset \mathcal{LD}(S_i)$ implies $\mathcal{LD}(S_j) \subseteq \mathcal{LD}(S_i)$. In fact, $\mathcal{LD}(S_j) = \mathcal{LD}(S_i)$.

The second key property also generalizes: if the LDOI of a given node state set contains the set itself, then the LDOI of the set contains at least one stable motif. The intuition and proof is similar to the case of a single node state. Taking the same example, consider $S_i = \{\sim n_3\}$ and $S_j = \{n_2, n_4\}$, note that both $S_i \subset \mathcal{LD}(S_i)$ and $S_j \subset \mathcal{LD}(S_j)$, this implies that both $\mathcal{LD}(S_i)$ and $\mathcal{LD}(S_j)$ contain a stable motif, which is SM_2 in this case.

Following these examples, we define the core of a stable motif to be a minimal subset of the stable motif whose LDOI contains the stable motif. Here by minimal we mean that no true subset of the core of the stable motif will contain the entire stable motif. The core of a stable motif can be a single node or more than one node. For example, as shown in **Figure 1C** $\sim n_3$ is a single-node core of the stable motif $SM_2 = \{n_0, \sim n_1, n_2, \sim n_3, n_4\}$. $\{n_2, n_4\}$ is another core of the same stable motif as $SM_2 \not\subset \mathcal{LD}(n_2)$, $SM_2 \not\subset \mathcal{LD}(n_4)$, and $SM_2 \subseteq \mathcal{LD}(\{n_2, n_4\})$.

We also define a driver node (set) of the stable motif to be a node (set) whose DOI contains the entire stable motif. The driver node (set) can be inside the stable motif, in which case it is the core of the stable motif; it can also be an upstream node that is sufficient to activate (the core of) the stable motif. We note that stabilization of a stable motif does not require the sustained state of a driver node, that is, oscillations can also lead to the stabilization of a stable motif. An example of this behavior was shown in **Figure 2B**: node B, which constitutes a self-sustaining stable motif, can stabilize by a single instance of A = 1, regardless of the fact that the negative self-regulation of A makes it oscillate.

2.7. Target Control Algorithm

Now that we have equipped ourselves with the tool of LDOI to find the long term effect of a sustained intervention, we can formulate the target control problem as the identification of a node set S^* whose LDOI contains the target node state set, i.e., $\mathcal{LD}(S^*) \supseteq Target$. This problem can be framed as a planning search problem (Russell and Norvig, 2003). We start with a null set whose LDOI is also null. We repeatedly add a new node to the set until the LDOI of this set contains the target node state set. We use LDOI instead of DOI for this purpose because identification of the DOI is a computationally more difficult problem. Our current solution using LDOI sets a tight upper bound for the optimal solution for the target control problem as $\mathcal{D}(S^*) \supseteq \mathcal{LD}(S^*) \supseteq Target$.

Previous work in the target control of Boolean models has focused on full enumeration of the solutions for the target control problem (Klamt et al., 2006; Samaga et al., 2010), which can be used to identify the solutions that involve combinations of a small number of nodes but is not generally viable because of combinatorial explosion. In our work, we use a complementary approach to avoid a full state space search in this combinatorial search problem. We apply a random heuristic algorithm called the greedy randomized adaptive search procedure (GRASP) (Pardalos et al., 1998; Festa et al., 2001). The pseudocode is described in Algorithm (**Tables 1**, **2**). The algorithm consists of two main phases. The first phase is the construction of a greedy randomized solution and the second phase is a local search to remove any redundancy of the solution.

Algorithm 1 GRASP algorithm for Target Control Problem

```
1:  procedure GRASP(G_expanded,Target,max_itr)
2:      solutions ← List()
3:      for index ← 1, max_itr do
4:          solution ←ConstructGreedyRandomizedSolution
                                        (G_expanded, Target)
5:          solution ←LocalSearch(G_expanded, Target, solution)
6:          if solution then
7:              Solutions.append(solution)
8:          end if
9:      end for
10:     return solutions
11: end procedure
```

Algorithm 2 Algorithm for constructing a greedy randomized solution

```
1:  procedure CONSTRUCTGREEDYRANDOMIZEDSOLUTION
                                        (G_expanded, Target)
2:      solution ← Set()
3:      α ← random(0, 1)
4:      candidates ←Construct_Initial_Candidates(G_expanded,
                                                  Target)
5:      G(v)       ←Construct_Greedy_Functions(G_expanded,
                                                  candidates)
6:      while candidates do
7:          RCL ←MakeRCL(candidates, G(v), α)
8:          s ←Select_Candidate(RCL)
9:          solution ← solution ∪ {s}
10:         if Target ⊂ LDOI(solution) then
11:             return solution
12:         end if
13:         Update_Candidates(candidates)
14:     end while
15:     return Set()
16: end procedure
```

In the first phase, we first generate an initial candidate list (line 4 in Algorithm 2). In the simplest case, the initial candidate list is all the non-composite nodes of the expanded network except

the nodes in the target set and their negation, both of which are ineligible for control. One can also be more selective to adapt to the specific needs of controlling biological systems. For example, we can forbid the use of certain nodes or node states when constructing the initial candidate list, to incorporate the fact that certain chemical species are harder or even unrealistic to control. Thus these nodes/chemical species will never appear in the final solution since they are not in the initial candidate list.

Then, we begin the procedure of iteratively adding nodes to the trial solution set (which is initially empty) and evaluating whether the LDOI of the trial solution set covers the target set. We form a restricted candidate list (RCL, line 7 in Algorithm 2) based on a greedy measure $G(v)$ defined for each candidate node v in the candidate list (line 5 in Algorithm 2). A greedy function is a heuristic score to estimate whether this node should be included in the solution. We evaluated five choices of $G(v)$, as described at the end of this section and in section 3.1. We determine the minimum score $G_{min} = min_{v\in V}\ G(v)$ and maximum score $G_{max} = max_{v\in V}\ G(v)$ among the heuristic scores of all the nodes. Then we use a previously generated random number α from a uniform distribution between 0 and 1 to set a passing score for the RCL as $G_{pass} = G_{min} + \alpha \cdot (G_{max} - G_{min})$. Then the RCL consist of nodes whose greedy function is no less than the passing score, i.e., $RCL = \{v \in V | G(v) \geq G_{pass}\}$. This procedure of generating RCL is summarized in Supplementary Material section 1.3.

Next we randomly pick a node from the RCL and add it to the current trial solution (line 8 and 9 in Algorithm 2). The trial solution is used as the *source* node set of the LDOI algorithm (whose pseudocode is presented in Supplementary Material section 1.1). If the target set is contained in the set of nodes returned by the LDOI algorithm, we end the first phase and start the second phase (local search procedure) with this candidate solution (line 10 and 11 in Algorithm 2). Otherwise, we update the candidate node set and start the next iteration toward adding another node from the RCL to the trial solution set. We update the candidate node set by removing the previously added node, its negation and any node in the LDOI of the current trial solution (line 13 in Algorithm 2). We do this latter exclusion because these nodes will stabilize because of the current trial solution, and it is useless to add any stabilized state to the trial solution. We repeat the whole procedure including selecting a node randomly from the candidate set as long as there are still candidate nodes (line 6 in Algorithm 2). We return an empty set if we do not find a solution (line 15 in Algorithm 2).

In the second phase (see the pseudocode in Supplementary Material section 1.2), we start with a candidate solution that covers the target set. We randomize the order of nodes in the candidate solution and then iteratively attempt to remove each node. If after removing this node the LDOI of the modified solution still covers the target set, then we replace the candidate solution with the modified solution. Thus after one iteration of traversing all the nodes, we obtain a final solution. At worst, no node is removed from the set and the final solution is the same as the candidate solution. The randomness in the removal order provides a possibility for obtaining different minimal solutions from the same candidate solution.

In this random heuristic algorithm, we introduce two aspects of randomness in the construction phase, one is the randomness of the passing score by a different α for each iteration of solution generation process (line 3 in Algorithm 1) and another is the random selection of a node each time from the RCL inside each solution generation process (line 8 in Algorithm 2). These techniques help strike a balance between the bias of a greedy function and exploring the whole node state space (Pardalos et al., 1998; Festa et al., 2001). An efficient greedy function/heuristic score is important to guide the search procedure toward the subspace with the optimal solution. However, a universally efficient greedy function may not exist; rather, the efficiency of a greedy function may depend on the specific network structure and target set. We have implemented five choices of greedy functions $G(v)$ for a given node state (equivalently, non-composite node of the expanded network): score 1 is the size of the LDOI of that node state (denoted as $|LDOI|$); score 2 is the size of the set of composite nodes which are nearest neighbors of the LDOI of that node state (denoted as $|Comp\_LDOI|$); score 3 is a linear combination of the previous two measures with equal weight (denoted as *Scores*_1+2), and score 4 and 5 as the size of the LDOI of that node state with penalty if the LDOI contains a node that is the negation of a node in the target set (denoted as $|LDOI|\_Pen1$ and $|LDOI|\_Pen2$). The penalty can be implemented by multiplying this score by -1 (score 4) or by decreasing this score by the size of the largest LDOI among all node states (score 5); both of these implementations ensure that this score becomes non-positive. A python implementation of the target control algorithm is available at https://github.com/yanggangthu/BooleanDOI.

2.8. Computational Complexity of the Target Control Algorithm

The time complexity of calculating the LDOI of any set is bounded by $O(N_{ex} + E_{ex})$, where N_{ex} is the number of nodes and E_{ex} is the number of edges of the expanded network. For each non-composite node in the network, we initially calculate its LDOI and the value of its greedy function, with time complexity $O(N(N_{ex} + E_{ex}))$, where N is the number of nodes in the original network. We then cache these results to improve the performance of the GRASP algorithm. In the first phase of the GRASP algorithm, we run at most N iterations and we need to calculate the LDOI of the trial solution in each iteration, thus the time complexity is bounded by $O(N(N_{ex} + E_{ex}))$. In the second phase, the time complexity is also bounded by $O(N(N_{ex} + E_{ex}))$ as we need to go through each node, bounded by $O(N)$ as a crude estimate, delete the node from the solution and check the modified solution's LDOI, which is $O(N_{ex} + E_{ex})$. The Boolean regulatory functions of biological network models are often nested canalizing rules (Kauffman et al., 2003; Li et al., 2013), thus for each node with k regulators there are at most k newly generated composite nodes in the expanded network, as well as two corresponding non-composite nodes; each of these nodes have at most k regulators. Thus N_{ex} is bounded by $O(\bar{k}N)$, and E_{ex} is bounded by $O(\overline{k^2}N)$. Biological networks are sparse, with an average node in-degree $1 < \bar{k} < 3$ (Newman, 2010).

Thus the complexity of the target control algorithm applied to biological network models is $O(\overline{k^2}N^2) \sim O(N^2)$ for a well-behaved degree distribution in the sparse limit and bounded by $O(N^3)$ for an extremely skewed degree distribution in the sparse limit. Different iterations of the solution generation process (line 3 in Algorithm 1) can be easily parallelized as each iteration is independent. The space complexity of BFS search on the expanded graph is bounded by $O(N_{ex})$, and the space complexity of the entire procedure is bounded by time complexity times the storage space of LDOI of a node set, which is bounded by $O(N_{ex})$.

2.9. Damage Mitigation as Target Control

We can generalize the target control algorithm to solve a damage mitigation problem. Consider a Boolean network that has two steady states, one corresponding to the normal state of the system and the other corresponding to a disease state. The system is currently in the normal steady state, but damage to a node, which causes it to stabilize in the opposite state, will lead the system to the disease steady state without any intervention. Under such conditions, previous research has proposed modifying the network topology (as soon as possible, or preventatively) to block the propagation of damage (Yang et al., 2016). Here we are interested in designing a damage mitigation strategy to bring the system back to an attractor similar to the normal steady state in the sense that a subset of nodes are in the same state as their states in the normal steady state. This problem is almost the same as the target control problem except that we need to take the permanent damage into consideration. There are two ways of incorporating this. First, we treat this permanent damage as an initial condition and apply network reduction to the system. However, this risks reducing a significant fraction of the nodes in the network, including the target nodes we are interested in. Second, we can apply our GRASP algorithm as above while initializing the solution with the damaged node state(s) and forbidding the damaged node state to be removed in the local search phase in GRASP algorithm. This means that we include the damage as part of the intervention. When the LDOI of the node state set containing the damage effect covers the target set, the target nodes will stabilize in their desired states after a finite number of time steps under all initial conditions of the subspace of the damaged network. We note that we only need to do this when the damage is a permanent one; when the damage is temporary (i.e., when the node is allowed to go back to its original state), this can be treated as a different initial condition for the target control problem and we can still apply our GRASP algorithm to solve it as DOI/LDOI is robust to any initial condition by definition.

3. RESULTS

3.1. Application to Ensembles of Random Boolean Networks

We tested the two proposed properties of the LDOI and the target control algorithm on different random Boolean network ensembles. Specifically, we generated an ensemble of 1000 Erdős Rényi random graphs (Newman, 2010) (using the gnm_random_graph() function of NetworkX; Hagberg et al., 2008), with size ranging from 15 to 50 nodes and average in-degree ranging from 1 to 2 . The Boolean regulatory functions of the random ensemble are required to be effective (irreducible) Boolean functions (Zertuche, 2009) to be consistent with the generated topology, or nested canalizing functions to simulate biological systems. (A nested canalizing Boolean function with k inputs can be generated by determining two sequences, the input sequence $(I_1, I_2, \cdots, I_k)$ and the output sequence $(O_1, O_2, \cdots, O_k)$, where I_i or O_i is either 0 or 1. The output o as a function of input configuration $(i_1, \cdots, i_k)$ is thus determined through the hierarchy $o = O_1$ if $i_1 = I_1$; $o = O_2$ if $i_1 \neq I_1$ and $i_2 = I_2$; $\cdots$; $o = O_k$ if $i_1 \neq I_1$, $\cdots$, $i_{k-1} \neq I_{k-1}$, $i_k = I_k$; $o = NOT\ O_k$ if $i_1 \neq I_1$, $\cdots$, $i_{k-1} \neq I_{k-1}$, $i_k \neq I_k$.) We have successfully tested and validated the two properties for the LDOI of each node in the generated networks. We also tested and validated the properties of the LDOI of node sets of size up to 3∼7 depending on the specific network (as the complexity of testing the property grows faster than N^k for $k << N$, where N is the network size and k is the node set size).

With respect to testing the target control algorithm, we generate 50 random target sets with size 2 or 3 for each random network. We calculate the average number of generated solutions for each pair formed by a target set and a network. As shown in **Table 1**, the average number of solutions is significantly high, between 10 and 40 for ensembles with nested canalizing functions and between 25 and 70 for ensembles with effective Boolean functions. The dominance of the canalizing variables in determining certain outcomes tends to yield sparser expanded networks than non-canalizing functions, and fewer effective interventions. This is reflected in the smaller number of solutions in the ensembles with nested canalizing functions compared to the ensembles with effective Boolean functions. As shown in Figure S1, the average time for finding solutions for a target set (through 500 iterations) is 100 s or less for networks with 15–35 nodes and 20–60 edges. As expected, the runtime increases with the number of nodes and edges, reaching 600 s for 50 nodes and 100 edges. The relatively slow increase and practical runtime suggest that our algorithm is effective for logical models of biological systems.

It is not always possible to find a solution for a specific target set for a network, especially when the Boolean network model

TABLE 1 | Mean number of solutions found for each target set and random network pair for 50 target sets and 1,000 networks.

Custom score index and notation	1 *LDOI*	2 *Comp_LDOI*	3 *Scores_1+2*	4 *LDOI_Pen1*	5 *LDOI_Pen2*
Nested canalizing rules	12.29	31.66	12.30	31.21	40.64
Effective boolean rules	26.21	61.94	26.22	57.08	66.91

Half of 50 target sets have size two and the other half is of size three; none of them contain source nodes. The 2nd to 6th columns correspond to different custom score (greedy function) indexes and notations, which are described in the last paragraph in section 2.7. The second and third row corresponds to the random network ensemble with nested canalizing rules and effective Boolean rules, respectively.

does not have a (partial) fixed point type of attractor (i.e., if all nodes oscillate in the attractor) or when the desired target state set consists of node states that are part of different attractors, which conflict with each other. Another case where target control is impossible is when the target set is not reachable from the rest of the network. In the simulations of the two ensembles mentioned above, we verified that we are able to find a solution for more than 99.5% of the target sets when the target set satisfies two criteria: (i) it is a subset of a (partial) fixed point and (ii) the targets in this set are accessible from nodes outside of this set in the original network (that is, the targets do not consist of source nodes only and do not form a motif without any incoming edges). Note that there can be counter-examples where satisfying these criteria is not sufficient to find a solution. For example, in **Figures 2A,B**, there are no solutions for the target set $\{\sim B, \sim C\}$ as the remaining nodes are not enough to activate the composite node in **Figure 2B**. However, the probability of such situations is small in both random ensembles with moderate size and real biological network. Moreover, the fact that one cannot find a solution through our GRASP algorithm for the target control problem often indicates that the target set is not a reasonable target. It is likely that one would not be able to find a solution in such situation even with a whole state space search.

We also test the performance of different heuristic functions for the target control problem. We calculate the average number of generated solutions for each pair formed by a target set and a network. As shown in **Table 1**, greedy functions with a penalty for containing the negation of a node state included in the target set (score index 4 and 5) consistently perform better than the greedy functions directly using the size of the LDOI (score index 1 and 3). The intuition behind this is that it is more efficient to choose from those nodes whose DOI does not contain a conflict with the target. The second greedy function ($|Comp\_LDOI|$) also performs quite well.

3.2. Biological Examples

We applied our methodology on four Boolean models of signal transduction networks. The four Boolean models are freely available on GitHub (https://github.com/yanggangthu/BooleanDOI) in SBML Qual format and in our custom format. In the following we demonstrate our algorithm on two of these, the epithelial-to-mesenchymal transition (EMT) network and the PI3K mutant ER+ breast cancer network. The results on the ABA induced stomatal closure network and the T-LGL leukemia network are shown in Supplementary Materials sections 3.3, 3.4. **Table 2** summarizes representative interventions found with our algorithm and compares them to the results of the most relevant previous analysis of these four biological network models. In Supplementary Data Sheet 1 we include the LDOI of each single node in the four models analyzed.

3.2.1. EMT Network

EMT is a cell fate change involved in embryonic development, which can be reactivated during cancer metastasis (Steinway et al., 2014). During EMT, epithelial cells lose their original adhesive property, and become mesenchymal cells which leave their primary site, invade neighboring tissue, and migrate to distant sites. A Boolean network model of EMT in the context of hepatocellular carcinoma invasion has been established by Steinway et al. (2014). Several predictions of this model were validated experimentally (Steinway et al., 2014, 2015). The EMT network has 70 nodes and 135 edges. The adhesion factor E-cadherin is the sink node; its OFF state indicates the transition to a mesenchymal state. The network has a normal (epithelial) steady state and an abnormal (mesenchymal) steady state. (See details in Supplementary Materials section 3.1). In **Figure 3** we show a simplified version of the EMT network; our analyses were done on the full network.

Previous research on this network has indicated that sustained activation of TGFβ signal can trigger EMT through the

TABLE 2 | Summary of representative target control solutions found by our algorithm for four biological network models.

Model	Target state(s)	Representative interventions found	Previous results from dynamic analysis	
			EMT-blocking	**E cell inducing**
EMT network	∼EMT	β-catenin_memb = 1, SNAI1 = 0	SNAI = 0	β-catenin_memb = 1, SMAD = RAS = 0, SNAI1 = GLI = 0
		β-catenin_memb = 1, SMAD = RAS = 0	SMAD = 0, RAS = 0	
	∼EMT, ∼MEK	β-catenin_memb = 1, SNAI1 = RAS = 0	Not studied previously	
		β-catenin_memb = 1, miR200 = RKIP = 1, RAS = 0		
Breast cancer network	Apoptosis = 2, Proliferation = 0	PI3K = 0, ESR1 = 0	PI3K = 0, ESR1 = 0	
	Apoptosis <2, Proliferation >2 when PI3K = 0	PI3K = 0, ESR1 = 1	PI3K=0, ESR1 = 1	
ABA induced closure network	Closure = 1 when ABA = 0	Ca_c^{2+} = ROS = 1	ROS = 1	
	H_2O efflux when ABA = 0	K^+ efflux =1, SLAC1 = ROS = 1	Not studied previously	
T-LGL leukemia network	Apoptosis = 1	S1P = 0, RAS = 1	S1P = 0	

The first column indicates the relevant network model, the second lists one or two target states we considered, the third column presents the intervention set obtained by our algorithm, and the fourth column indicates the most relevant results of previous analysis of these network models. The previous analysis considered a unique initial state or a restricted family of initial states (in the case of the ABA network). The interventions found by our algorithm will be successful regardless of the initial state.

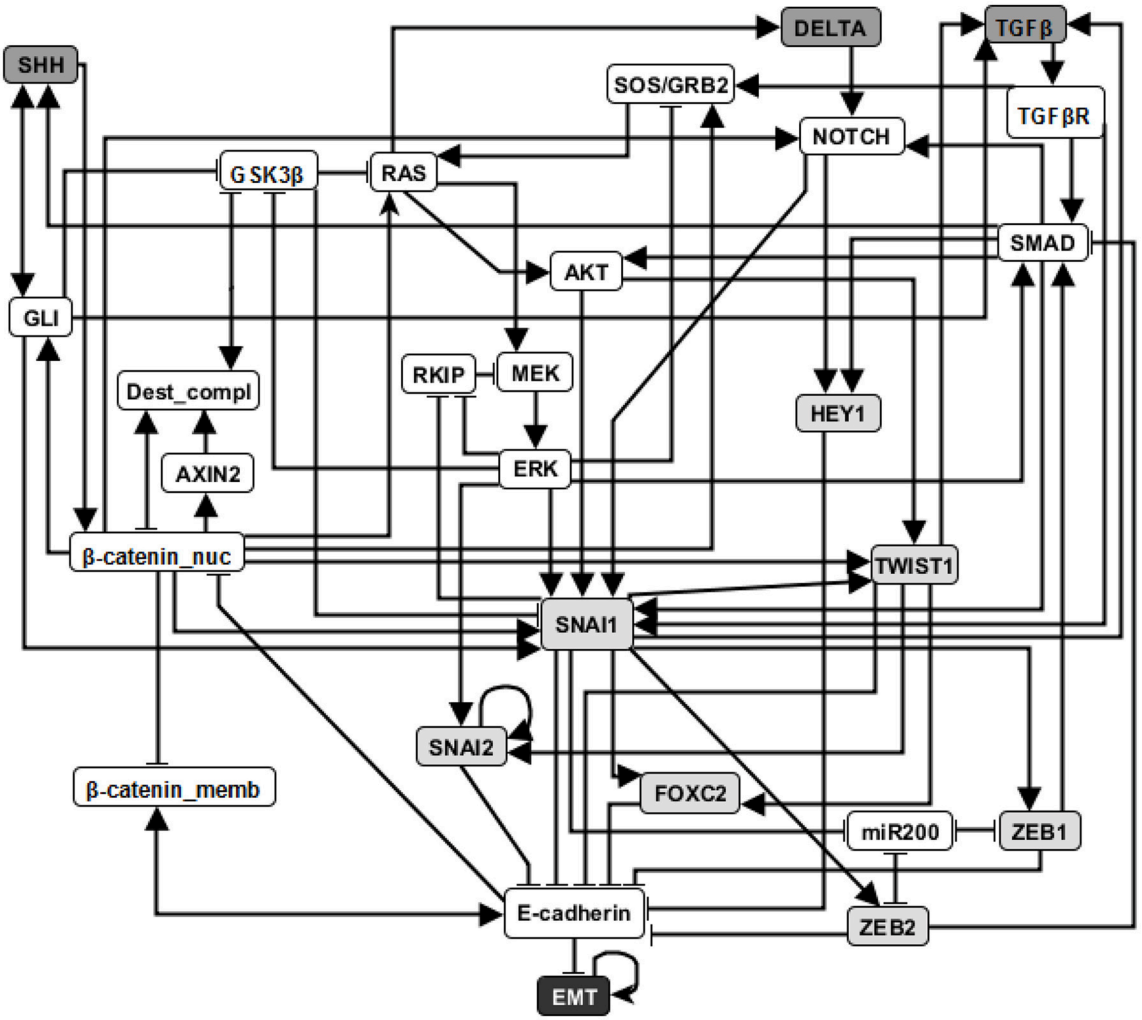


FIGURE 3 | An illustration of the EMT network. Attractor-preserving network reduction (Saadatpour et al., 2013) was applied to better focus attention on the most relevant nodes. Specifically, source nodes that represent input signals that are absent in the studied context are removed and each node with one input or one output is absorbed into its input or output, respectively. Nodes with light gray background are direct regulators of E-cadherin and nodes with dark gray background represent external signaling molecules. Edges ending with an arrow represent positive regulation while edges end with a flat bar represent negative regulation. See more details in Supplementary Material section 3.1.

activation of eight stable motifs (Steinway et al., 2015). In addition, stabilization of any of these stable motifs can drive EMT. Our analysis of the LDOI of each node state indicates that any of 60 node states (out of 138 node states for the 69 nodes) can lead to EMT, including the previously established EMT drivers. Moreover, 43 node states (nodes of the expanded network) have the same LDOI, which contains 48 node states, including EMT = 1 (see Supplementary Data Sheet 1). Each of these 43 node states is either the core of one or more of the eight stable motifs, or an external driver of one or more of the eight stable motifs. Thus the EMT outcome and the mesenchymal steady state has a large basin of attraction. As we are more interested in designing therapeutic strategies to block the epithelial to mesenchymal transition, we set the negation of EMT as a target. Previous analysis indicated that when considering an initial epithelial state and turning on the TGFβ signal, the knockout of any of the transcription factors that downregulate E-cadherin (i.e., knockout of SNAI1, SNAI2, FOXC2, TWIST1, ZEB1, ZEB2, HEY1) or multiple double node knockout combinations (knockout of SMAD and one of RAS, CSL, DELTA, NOTCH, NOTCH_ic, SOS/GRB2) are effective in blocking EMT (i.e., leading to E-cadherin=ON). The effectiveness of transcription factor knockout had been established in the literature; unfortunately these transcription factors cannot be targeted with existing drugs. Several double knockout combinations were validated experimentally in (Steinway et al., 2015) and are more amenable to drug targeting.

For EMT as target, our target control algorithm gives 7 two-node solutions (activation of β-catenin_memb and knockout of

any of SNAI1, SNAI2, FOXC2, TWIST1, ZEB1, ZEB2, HEY1) and 5 three-node solutions (activation of β-catenin_memb, knockout of SMAD and knockout of any of RAS, CSL, DELTA, NOTCH, and NOTCH_ic). The main difference between the target control solution and the previously found EMT-blocking single and double knockout interventions is that our target control solution includes the additional control of β-catenin_memb. To understand this difference, we note that EMT is in the LDOI of TFGβ, however, EMT is not in the LDOI of the set consisting of TGFβ together with any of the previously found EMT-blocking knockout interventions. This indicates that the knockout intervention is effective in the sense that it can block the process of reaching EMT. However, ∼EMT is also not in the LDOI of the set of TGFβ together with any knockout intervention. The knockout intervention is effective when the initial condition is the epithelial steady state, however the knockout intervention does not block EMT for all initial conditions. The target control algorithm, which can block EMT for all initial conditions, requires one more node (β-catenin_memb) in the target control solution. In fact, treating this problem as a damage mitigation problem, where the damage is sustained activation of TGFβ, we verify that EMT is in the LDOI of TGFβ together with any of the target control solutions.

As established in previous results, the single node EMT-blocking knockouts do not lead back to an epithelial state but rather to hybrid epithelial or mesenchymal steady states (Steinway et al., 2015). The hybrid epithelial steady state has certain epithelial features, e.g., E-cadherin and β-catenin_memb are activated, and also some mesenchymal features, e.g., MEK, ERK, and SNAI1 are activated. The hybrid mesenchymal steady state demonstrates the opposite features compared to the epithelial steady state. A good target set to avoid reaching such a hybrid state (which is likely pathological and may even be a worse outcome as the mesenchymal state) would be {∼EMT, ∼MEK} (Steinway et al., 2015). The minimum solution found involves controlling three nodes: activation of β-catenin_memb, inhibition of SNAI1, inhibition of RAS or RAF. We also find a four-node intervention that does not involve ERK and SNAI1: activation of β-catenin_memb, miR200 and RKIP, and also inhibition of RAS. If the target set is {∼EMT, ∼MEK, ∼SNAI1}, the minimum solution size is found to be six.

Stable motif control indicates that control of at least five nodes is needed to drive any initial state (including the mesenchymal state) to the epithelial state (see Supplementary Table 3 of Steinway et al., 2015) Although the control goal is different, one can still see the connection between our target control solution for the target ∼EMT and the stable motif control solution (to drive the system to the epithelial state). Specifically, they both require activation of β-catenin_memb. Knockout of SNAI1 , knockout of TWIST1, or knockout of SMAD and RAS, as one of the target control solutions, also appear as a part of stable motif control solution that does not require control of TGFβ or TGFβR.

These results demonstrate both the accuracy and effectiveness of our target control algorithm. The solutions found through 1,000 iterations are comprehensive (comparable to the solution found through a systematic search of knockout pairs). Our algorithm indicates intervention sites that are close to the target but also sites that are further away (e.g., SMAD). This diversity enables the selection of the most practical interventions.

3.2.2. Breast Cancer Network

In 2017, Zañudo et al. established a discrete dynamical model of the signal transduction processes involved in the PI3K mutant, estrogen receptor positive (ER+) breast cancer, as shown in **Figure 4** (Gómez Tejeda Zañudo et al., 2017). The model includes 58 nodes, which correspond to proteins, transcripts, drugs, and two cellular outcomes, apoptosis (programmed cell death) and proliferation (cell cycle progression). A fraction of the nodes (16), including the outcome nodes, are characterized by multiple levels, which is implemented by additional virtual nodes, e.g., apoptosis2 corresponds to level 2 of apoptosis, which has a more stringent regulatory function than apoptosis1 (level 1 of apoptosis). This network as implemented is essentially a Boolean network because all the regulatory functions are Boolean (Gómez Tejeda Zañudo et al., 2017). The network model successfully captures the key role of the PI3K/AKT/mTOR signaling pathway in determining the pathological proliferation and survival of cancer cells. In untreated simulated cancers cells, PI3K, MAPK, AKT, mTORC1, and ER signaling are active, leading to high level of proliferation and lack of apoptosis. The network model successfully captures the effectiveness of PI3K inhibiting drugs in leading to low level of proliferation and high level of apoptosis (Gómez Tejeda Zañudo et al., 2017). Through extensive simulations, the network model confirms known drug resistance mechanisms, i.e., additional mutations or other dysregulations that lead to the loss of effectiveness of PI3K-inhibiting drugs. It also predicts new possible resistance mechanisms and the degree of survivability under different resistance mechanisms (Gómez Tejeda Zañudo et al., 2017).

Similar insights can be drawn by LDOI analysis and applying the target control algorithm to the discrete dynamical network model without doing dynamical simulations, which demonstrate the rich information contained in the network topology and logic and the effectiveness of our control methodology. We obtained a (relatively large) reduced network by considering the system under the relevant initial condition of PI3K mutant, ER+ cancerous state, while keeping the seven drugs as source nodes (see details in Supplementary Material section 3.2). The five node states with the highest LDOI are Fulvestrant, ∼ESR1 (which both mean the inhibition of the estrogen receptor) and Alpelisib, ∼PI3K, ∼PIP3 (which all mean the inhibition of PI3K). The LDOI of these four nodes is very similar and includes 18 node states, including a high level of apoptosis (Apoptosis = 2), and a reduction in proliferation. Other drugs or node inhibitions yield subsets of the largest LDOI (see Supplementary Data Sheet 1). These results are consistent with, and yield further insight into the current knowledge on the effect of drugs in this network. If we now set the target to be high level of apoptosis and no proliferation, i.e., Target = {Apoptosis2, ∼Proliferation}, the algorithm gives multiple two-node interventions as minimal interventions, these consists of either of {∼PI3K, ∼PIP3} and inhibition of any node in the MYC-CDK4/6 axis of cell-cycle regulation, i.e., {∼ESR1, ∼ER_transcription, ∼MYC, ∼CDK46,

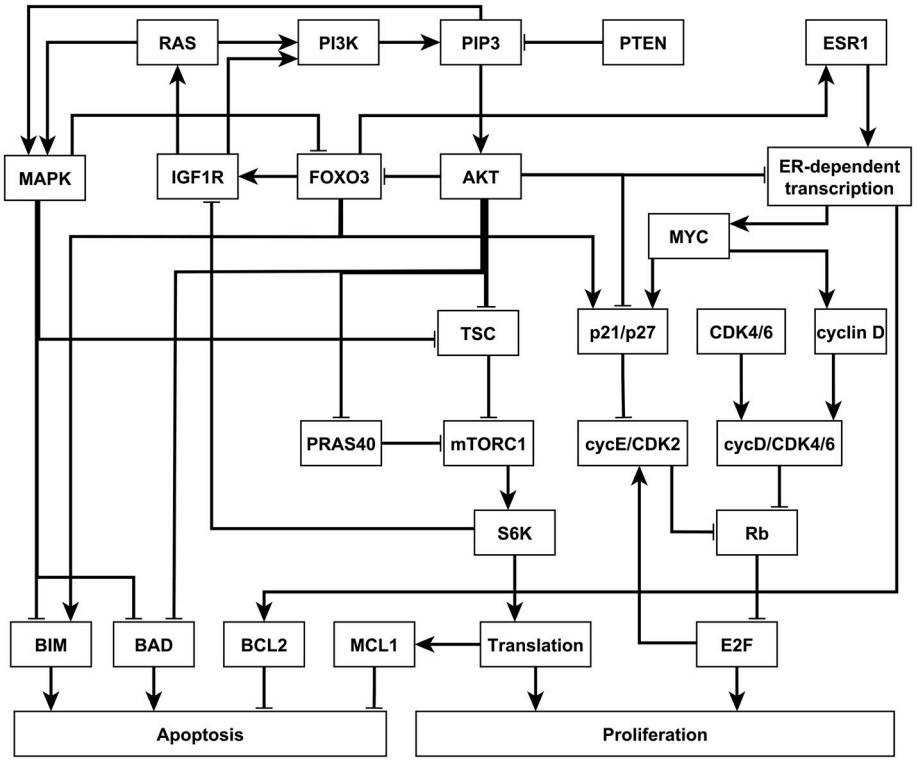


FIGURE 4 | An illustration of the PI3K mutant, ER+ breast cancer network. Attractor-preserving network reduction was applied to focus on the nodes most relevant to our analysis. Nodes are colored according to the signaling pathway that they participate in. Edges ending with an arrow represent positive regulation while edges ending with a hollow diamond represent negative regulation. See more details in Supplementary Material section 3.2.

∼cyclinD, ∼cycD_CDK46, ∼Rb, ∼E2F}. There are several drugs that can target these nodes. For example, Alpelisib is a PI3K inhibitor, Fulvestrant is a ESR1 inhibitor and Palbociclib is a CDK4/6 inhibitor. This result is consistent with the results found in the (Gómez Tejeda Zañudo et al., 2017): inhibition of PI3K leads to an increase in ER transcriptional regulatory activity, leading to a decrease in proliferation, and simultaneous PI3K and ER inhibition has a synergistic effect in completely blocking proliferation and maintaining a high level of apoptotic activity. If PI3K inhibitor or PIP3 inhibitor is not allowed to be used, the algorithm finds three node solutions involving an AKT inhibitor (e.g., Ipatasertib), MAPK inhibitor (e.g., Trametinib) and inhibition of any node from the MYC-CDK4/6 axis of cell-cycle regulation. In other words, inhibition of AKT together with MAPK provides a similar functionality with inhibition of PI3K. One can also use the LDOI to identify possible drug resistance mechanisms, i.e., perturbations that make PI3K inhibition less effective. As $\{\text{Apoptosis2},\sim\text{Proliferation4}\} \subset \mathcal{LD}(\sim\text{PI3K})$, we simply go through all possible two-node interventions containing PI3K inhibitor and screen out those interventions whose LDOI either does not contain Apoptosis2 or contain Proliferation3 or higher level (Proliferation4). We reproduce most of the potential drug resistance mechanism to PI3K inhibitors indicated in Table 3 of Gómez Tejeda Zañudo et al. (2017).

4. DISCUSSION

In summary, we have developed the new measures DOI and LDOI to describe the long-term effect of a sustained intervention. We have applied these measures to find solutions to the target control problem in logical network models. This work takes a step forward toward practical control of real biological systems, as illustrated by the applications presented here. The target control solutions we find recover previous predicted interventions obtained by other methods (dynamic simulations and stable motif analysis). As several of these previous predictions are validated experimentally, this agreement also serves as validation of our target control solutions. Notably, by generating a large number of valid target control solutions, we are going significantly beyond previous results (see **Table 2**). The multitude of predicted target control interventions allows their filtering according to biological or technological considerations.

Here we assumed the existence of a discrete dynamical model. As there are significant uncertainties in the existing models due to the scarcity of experimental information, we estimate the sensitivity of the LDOI measure to the incompleteness of the dynamical model. As the primary way of obtaining causal information that can be used in a logical model is to perform knockout experiments, the predominant causal information indicates a node as being necessary for the activation of another

node. For example, if the knockout of either of two regulators A or B leads to a decrease in the activity of target C, we would infer that the logical rule for C is $C = A$ AND B. Suppose that there is a so far undetected regulator of C, which we denote by X. This X will likely also be necessary, which would maintain agreement with the previous observations, i.e., $C = A$ AND B AND X is the true rule. Consider the rule for the complementary node $\sim C = \sim A$ OR $\sim B$ in the case of the incomplete system versus the true rule $\sim C = \sim A$ OR $\sim B$ OR $\sim X$. We can see that the LDOI of any of $\sim A, \sim B, A, B$ will be robust to the addition of X. The LDOI of node X and $\sim X$ need to be established in the true system. The LDOI of node state set $\{A, B\}$ will be affected by this change. (However, LDOI of $\sim A$ and $\sim B$ will not change). Thus the size of the solution of the target control problem may increase due to this incomplete information. Due to the binary essence of the Boolean rule, missing a sufficient regulator (an extra OR rule) will give similar results.

The LDOI is closely related to previously introduced concepts and methods used to analyze Boolean models. In particular, the LDOI of a node state (or a set of node states) is mathematically equivalent to the three-valued logical steady state that results when these node states of interest are fixed (Klamt et al., 2006; Samaga et al., 2010) and is also equivalent to the set of node states that become stationary if using network reduction techniques after fixing the node(s) of interest in the appropriate state(s) (Bilke and Sjunnesson, 2001; Naldi et al., 2012; Saadatpour et al., 2013). The work presented here goes beyond previous work and identifies general properties of the LDOI of node states and their union (the first key property of the LDOI), and of the relation of the LDOI and stable motifs (the second key property of the LDOI).

The algorithm to identify the LDOI using the expanded network bears similarities with the algorithm in Samaga et al. (2010) and Klamt et al. (2006), which uses signed interaction hypergraphs to calculate logical steady states resulting from fixing node states. An important difference is that the expanded network assigns a complementary node to each node to denote the inactive state of a node, while the hypergraph representation instead assign signs to nodes and to composite nodes to keep track of their states. Although the LDOI obtained using either method is the same, we argue that the expanded network representation has several desirable properties that differentiate it, in particular, (i) it makes the interpretation of the LDOI more intuitive and the algorithm for calculating it purely graph-theoretical, i.e., a modified breadth-first search on the expanded network, (ii) it treats the active/inactive states equally (a reflection of the fact that a change of variables can redefine what an active/inactive state means), and (iii) it provides a natural way to generalize the LDOI from Boolean to discrete models by defining a virtual node for each allowed node state (e.g., if a node has 3 states we would have 3 virtual nodes: one denoting state 0, one denoting state 1, and one denoting state 2).

The DOI and LDOI is also related the concept of elementary signaling mode (ESM), originally defined as a minimal subgraph that can propagate a signal from a source node to an output node (Wang and Albert, 2011; Sun and Albert, 2016). An ESM on the expanded network is the generalization of a path on a usual directed network. Similarly, the LDOI of a node on the expanded network is analogous to a connected component reachable from a node on a usual directed network. In the same way a connected component reachable from node i consists of nodes that have a path starting from node i, the LDOI of a node consists of all the nodes included in any ESM that starts from that node. Recent work by Maheshwari and Albert (2017) developed a logic framework to identify causal relationships that are sufficient or necessary. This framework allows an alternative definition of the LDOI. The LDOI of the ON state of a node ($\tilde{\sigma}_i = 1$) includes all the nodes for which the node is a sufficient activator (these nodes will have $\tilde{\sigma}_j = 1$) or sufficient inhibitor (these nodes will have $\tilde{\sigma}_k = 0$). Similarly, the LDOI of the OFF state of a node includes all the nodes for which the node is a necessary activator (these nodes will have $\tilde{\sigma}_j = 0$) or necessary inhibitor (these nodes will have $\tilde{\sigma}_k = 1$).

An algorithm to construct ESMs through a backward search from an output node was presented in Wang et al. (2013); this algorithm can be adapted to find solutions of the target control problem of a single output. If we treat the output node as the root of a backward search, the set of nodes found in the ESM in each search depth (distance from the output node) can serve as a control solution. A truncation technique similar to ours needs to be applied to deal with inconsistent feed-forward or feed-back loops. This algorithm can be generalized to solve the target control problem of a target set by simultaneous search from each target node. We chose to transform the target control problem into a planning search problem; and it has been established that such a planning search problem can be solved in both a forward propagation and a backward propagation approach, or even a mixed approach (Russell and Norvig, 2003). It will be an interesting future work if such techniques can improve the efficiency of the algorithm.

This work points out interesting questions as future research directions. First, though evaluating the DOI of a node (set) is computationally hard, a better estimation of the DOI rather than the LDOI is desirable and can be used to reduce the size of the solution given by our current target control algorithm. Second, the requirement that the solution works for all initial conditions in the setup of the target control problem gives robust solutions, however it may be overly conservative for biological systems in certain applications, especially if one is certain about the relevant initial condition subspace. A semi-structural approach (without doing dynamical simulations) to solve the target control problem starting from a subspace of initial conditions are also desirable.

AUTHOR CONTRIBUTIONS

GY, JG, and RA designed research and methodology. GY and RA performed the analyses. GY, JG, and RA wrote the paper.

FUNDING

This work was supported by NSF grants PHY-1205840, IIS-1161007, PHY-1545832, the Stand Up to Cancer

Foundation, and a Stand Up to Cancer Foundation/The V Foundation Convergence Scholar Award (D2015-039) to JG.

ACKNOWLEDGMENTS

This research or portions of this research were conducted with Advanced CyberInfrastructure computational resources provided by The Institute for CyberScience at The Pennsylvania State University (http://ics.psu.edu).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphys.2018.00454/full#supplementary-material

REFERENCES

Abou-Jaoudé, W., Monteiro, P. T., Naldi, A., Grandclaudon, M., Soumelis, V., Chaouiya, C., et al. (2015). Model checking to assess t-helper cell plasticity. *Front. Bioeng. Biotechnol.* 2:86. doi: 10.3389/fbioe.2014.00086

Abou-Jaoudé, W., Traynard, P., Monteiro, P. T., Saez-Rodriguez, J., Helikar, T., Thieffry, D., et al. (2016). Logical modeling and dynamical analysis of cellular networks. *Front. Genet.* 7:94. doi: 10.3389/fgene.2016.00094

Albert, R., Acharya, B. R., Jeon, B. W., Zañudo, J. G. T., Zhu, M., Osman, K., et al. (2017). A new discrete dynamic model of ABA-induced stomatal closure predicts key feedback loops. *PLoS Biol.* 15:e2003451. doi: 10.1371/journal.pbio.2003451

Albert, R., and Othmer, H. G. (2003). The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*. *J. Theor. Biol.* 223, 1–18. doi: 10.1016/S0022-5193(03)00035-3

Alon, U. (2006). *An Introduction to Systems Biology: Design Principles of Biological Circuits, 1 Edn*. Chapman and Hall/CRC.

Azuma, S., Yoshida, T., and Sugie, T. (2014). "Structural monostability of activation-inhibition boolean networks," in *53rd IEEE Conference on Decision and Control*, 1521–1526.

Barabási, A.-L., and Pósfai, M. (2016). *Network Science*. Cambridge: Cambridge University Press.

Bilke, S., and Sjunnesson, F. (2001). Stability of the Kauffman model. *Phys. Rev. E* 65:016129. doi: 10.1103/PhysRevE.65.016129

Bloomingdale, P., Nguyen, V. A., Niu, J., and Mager, D. E. (2018). Boolean network modeling in systems pharmacology. *J. Pharmacokinet. Pharmacodyn.* 45, 159–180. doi: 10.1007/s10928-017-9567-4

Bornholdt, S. (2008). Boolean network models of cellular regulation: prospects and limitations. *J. R. Soc. Interface* 5(Suppl. 1), S85–S94. doi: 10.1098/rsif.2008.0132.focus

Cohen, D. P. A., Martignetti, L., Robine, S., Barillot, E., Zinovyev, A., and Calzone, L. (2015). Mathematical modelling of molecular pathways enabling tumour cell invasion and migration. *PLOS Comput. Biol.* 11:e1004571. doi: 10.1371/journal.pcbi.1004571

Cornelius, S. P., Kath, W. L., and Motter, A. E. (2013). Realistic control of network dynamics. *Nat. Commun.* 4:1942. doi: 10.1038/ncomms2939

Espinosa-Soto, C., Padilla-Longoria, P., and Alvarez-Buylla, E. R. (2004). A gene regulatory network model for cell-fate determination during *Arabidopsis thaliana* flower development that is robust and recovers experimental gene expression profiles. *Plant Cell* 16, 2923–2939. doi: 10.1105/tpc.104.021725

Festa, P., Pardalos, P. M., and Resende, M. G. C. (2001). Algorithm 815: Fortran subroutines for computing approximate solutions of feedback set problems using grasp. *ACM Trans. Math. Softw.* 27, 456–464. doi: 10.1145/504210.504214

Gao, J., Liu, Y.-Y., D'Souza, R. M., and Barabási, A.-L. (2014). Target control of complex networks. *Nat. Commun.* 5:5415. doi: 10.1038/ncomms6415

Glass, L. (1975). Classification of biological networks by their qualitative dynamics. *J. Theor. Biol.* 54, 85–107. doi: 10.1016/S0022-5193(75)80056-7

Glass, L., and Kauffman, S. A. (1973). The logical analysis of continuous, non-linear biochemical control networks. *J. Theor. Biol.* 39, 103–129.

Gómez Tejeda Zañudo, J., Scaltriti, M., and Albert, R. (2017). A network modeling approach to elucidate drug resistance mechanisms and predict combinatorial drug treatments in breast cancer. *Cancer Convergence* 1:5. doi: 10.1186/s41236-017-0007-6

Grieco, L., Calzone, L., Bernard-Pierrot, I., Radvanyi, F., Kahn-Perlès, B., and Thieffry, D. (2013). Integrative modelling of the influence of mapk network on cancer cell fate decision. *PLoS Comput. Biol.* 9:e1003286. doi: 10.1371/journal.pcbi.1003286.

Hagberg, A. A., Schult, D. A., and Swart, P. J. (2008). "Exploring network structure, dynamics, and function using networkx," in *Proceedings of the 7th Python in Science Conference*, eds G. Varoquaux, T. Vaught, and J. Millman (Pasadena, CA), 11–15.

Hopkins, A. L., and Groom, C. R. (2002). The druggable genome. *Nat. Rev. Drug Discov.* 1, 727–730. doi: 10.1038/nrd892

Iyengar, R., Zhao, S., Chung, S.-W., Mager, D. E., and Gallo, J. M. (2012). Merging systems biology with pharmacodynamics. *Sci. Trans. Med.* 4:126ps7. doi: 10.1126/scitranslmed.3003563

Kauffman, S. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.* 22, 437–467.

Kauffman, S., Peterson, C., Samuelsson, B., and Troein, C. (2003). Random boolean network models and the yeast transcriptional network. *Proc. Natl. Acad. Sci. U.S.A.* 100, 14796–14799. doi: 10.1073/pnas.2036429100

Khan, F. M., Marquardt, S., Gupta, S. K., Knoll, S., Schmitz, U., Spitschak, A., et al. (2017). Unraveling a tumor type-specific regulatory core underlying E2F1-mediated epithelial-mesenchymal transition to predict receptor protein signatures. *Nat. Commun.* 8:198. doi: 10.1038/s41467-017-00268-2

Kim, Y., Choi, S., Shin, D., and Cho, K.-H. (2017). Quantitative evaluation and reversion analysis of the attractor landscapes of an intracellular regulatory network for colorectal cancer. *BMC Syst. Biol.* 11:45. doi: 10.1186/s12918-017-0424-2

Klamt, S., Saez-Rodriguez, J., Lindquist, J. A., Simeoni, L., and Gilles, E. D. (2006). A methodology for the structural and functional analysis of signaling and regulatory networks. *BMC Bioinformatics* 7:56. doi: 10.1186/1471-2105-7-56

Klarner, H., and Siebert, H. (2015). Approximating attractors of boolean networks by iterative CTL model checking. *Front. Bioeng. Biotechnol.* 3:130. doi: 10.3389/fbioe.2015.00130

Klemm, K., and Bornholdt, S. (2005). Stable and unstable attractors in Boolean networks. *Phys. Rev. E* 72:055101. doi: 10.1103/PhysRevE.72.055101

Kraeutler, M. J., Soltis, A. R., and Saucerman, J. J. (2010). Modeling cardiac β-adrenergic signaling with normalized-hill differential equations: comparison with a biochemical model. *BMC Syst. Biol.* 4:157. doi: 10.1186/1752-0509-4-157

Laubenbacher, R., Hinkelmann, F., Murrugarra, D., and Veliz-Cuba, A. (2014). "Algebraic models and their use in systems biology," in *Discrete and Topological Models in Molecular Biology*, Natural Computing Series (Berlin; Heidelberg: Springer), 443–474.

Li, F., Long, T., Lu, Y., Ouyang, Q., and Tang, C. (2004). The yeast cell-cycle network is robustly designed. *Proc. Natl. Acad. Sci. U.S.A.* 101, 4781–4786. doi: 10.1073/pnas.0305937101

Li, Y., Adeyeye, J. O., Murrugarra, D., Aguilar, B., and Laubenbacher, R. (2013). Boolean nested canalizing functions: a comprehensive analysis. *Theor. Comp. Sci.* 481, 24–36. doi: 10.1016/j.tcs.2013.02.020

Lin, C.-T. (1974). Structural controllability. *IEEE Trans. Automatic Control* 19, 201–208.

Liu, Y.-Y., and Barabási, A.-L. (2016). Control principles of complex systems. *Rev. Mod. Phys.* 88:035006. doi: 10.1103/RevModPhys.88.035006

Liu, Y.-Y., Slotine, J.-J., and Barabási, A.-L. (2011). Controllability of complex networks. *Nature* 473, 167–173. doi: 10.1038/nature10011

Maheshwari, P., and Albert, R. (2017). A framework to find the logic backbone of a biological network. *BMC Syst. Biol.* 11:122. doi: 10.1186/s12918-017-0482-5

McCluskey, E. J. (1956). Minimization of Boolean functions. *Bell Syst. Tech. J.* 35, 1417–1444.

Méndez-López, L. F., Davila-Velderrain, J., Domínguez-Hüttinger, E., Enríquez-Olguín, C., Martínez-García, J. C., and Alvarez-Buylla, E. R. (2017). Gene regulatory network underlying the immortalization of epithelial cells. *BMC Syst. Biol.* 11:24. doi: 10.1186/s12918-017-0393-5

Mendoza, L. (2006). A network model for the control of the differentiation process in th cells. *Biosystems* 84, 101–114. doi: 10.1016/j.biosystems.2005.10.004

Miskov-Zivanov, N., Turner, M. S., Kane, L. P., Morel, P. A., and Faeder, J. R. (2013). The duration of T cell stimulation is a critical determinant of cell fate and plasticity. *Sci. Signal.* 6:ra97. doi: 10.1126/scisignal.2004217

Mochizuki, A., Fiedler, B., Kurosawa, G., and Saito, D. (2013). Dynamics and control at feedback vertex sets. II: a faithful monitor to determine the diversity of molecular activities in regulatory networks. *J. Theor. Biol.* 335, 130–146. doi: 10.1016/j.jtbi.2013.06.009

Morris, M. K., Saez-Rodriguez, J., Sorger, P. K., and Lauffenburger, D. A. (2010). Logic-based models for the analysis of cell signaling networks. *Biochemistry* 49, 3216–3224. doi: 10.1021/bi902202q

Motter, A. E. (2015). Networkcontrology. *Chaos* 25:097621. doi: 10.1063/1.4931570

Murrugarra, D., and Dimitrova, E. S. (2015). Molecular network control through boolean canalization. *EURASIP J. Bioinformatics Syst. Biol.* 2015:9. doi: 10.1186/s13637-015-0029-2

Murrugarra, D., Veliz-Cuba, A., Aguilar, B., and Laubenbacher, R. (2016). Identification of control targets in boolean molecular network models via computational algebra. *BMC Syst. Biol.* 10:94. doi: 10.1186/s12918-016-0332-x

Naldi, A., Carneiro, J., Chaouiya, C., and Thieffry, D. (2010). Diversity and plasticity of th cell types predicted from regulatory network modelling. *PLoS Comput. Biol.* 6:e1000912. doi: 10.1371/journal.pcbi.1000912

Naldi, A., Monteiro, P., and Chaouiya, C. (2012). "Efficient handling of large signalling-regulatory networks by focusing on their core control," in *Computational Methods in Systems Biology,* Lecture Notes in Computer Science, Vol. 7605, eds D. Gilbert and M. Heiner (Berlin; Heidelberg: Springer), 288–306.

Newman, M. E. J. (2010). *Networks: An Introduction*. Oxford, NY: Oxford University Press.

Nicholl, D. S. T. (2008). *An Introduction to Genetic Engineering, 3 Edn*. Cambridge, UK: Cambridge University Press.

Palsson, B. (2006). *Systems Biology: Properties of Reconstructed Networks*. Cambridge, NY: Cambridge University Press.

Papin, J. A., Hunter, T., Palsson, B. O., and Subramaniam, S. (2005). Reconstruction of cellular signalling networks and analysis of their properties. *Nat. Rev. Mol. Cell Biol.* 6, 99–111. doi: 10.1038/nrm1570

Pardalos, P. M., Qian, T., and Resende, M. G. (1998). A greedy randomized adaptive search procedure for the feedback vertex set problem. *J. Combinatorial Optim.* 2, 399–412.

Paulevé, L., Magnin, M., and Roux, O. (2012). Static analysis of biological regulatory networks dynamics using abstract interpretation. *Math. Struct. Comp. Sci.* 22, 651–685. doi: 10.1017/S0960129511000739

Paulevé, L., and Richard, A. (2012). Static analysis of boolean networks based on interaction graphs: a survey. *Electron. Notes Theor. Comput. Sci.* 284, 93–104. doi: 10.1016/j.entcs.2012.05.017

Russell, S. J., and Norvig, P. (2003). *Artificial Intelligence : A Modern Approach*. Upper Saddle River, NJ: Prentice Hall/Pearson Education.

Saadatpour, A., Albert, I., and Albert, R. (2010). Attractor analysis of asynchronous boolean models of signal transduction networks. *J. Theor. Biol.* 266, 641–656. doi: 10.1016/j.jtbi.2010.07.022

Saadatpour, A., and Albert, R. (2013). Boolean modeling of biological regulatory networks: A methodology tutorial. *Methods* 62, 3–12. doi: 10.1016/j.ymeth.2012.10.012

Saadatpour, A., Albert, R., and Reluga, T. C. (2013). A reduction method for boolean network models proven to conserve attractors. *SIAM J. Appl. Dyn. Syst.* 12, 1997–2011. doi: 10.1137/13090537X

Saez-Rodriguez, J., Simeoni, L., Lindquist, J. A., Hemenway, R., Bommhardt, U., Arndt, B., et al. (2007). A logical model provides insights into T cell receptor signaling. *PLoS Comput. Biol.* 3:e163. doi: 10.1371/journal.pcbi.0030163

Samaga, R., Kamp, A. V., and Klamt, S. (2010). Computing combinatorial intervention strategies and failure modes in signaling networks. *J. Comput. Biol.* 17, 39–53. doi: 10.1089/cmb.2009.0121

Shalem, O., Sanjana, N. E., Hartenian, E., Shi, X., Scott, D. A., Mikkelson, T., et al. (2014). Genome-scale CRISPR-cas9 knockout screening in human cells. *Science* 343, 84–87. doi: 10.1126/science.1247005

Steinway, S. N., Wang, R.-S., and Albert, R. (2016). "Discrete dynamic modeling: a network approach for systems pharmacology," in *Systems Pharmacology and Pharmacodynamics,* eds D. Mager and H. Kimko (Cham: Springer International Publishing), 81–103.

Steinway, S. N., Zañudo, J. G., Ding, W., Rountree, C. B., Feith, D. J., Loughran, T. P., et al. (2014). Network modeling of TGFβ signaling in Hepatocellular Carcinoma Epithelial-to-Mesenchymal Transition reveals joint sonic Hedgehog and Wnt pathway activation. *Cancer Res.* 74, 5963–5977. doi: 10.1158/0008-5472.CAN-14-0225

Steinway, S. N., Zañudo, J. G. T., Michel, P. J., Feith, D. J., Loughran, T. P., and Albert, R. (2015). Combinatorial interventions inhibit tgfb-driven epithelial-to-mesenchymal transition and support hybrid cellular phenotypes. *Npj Syst. Biol. Appl.* 1:15014. doi: 10.1038/npjsba.2015.14

Sun, Z., and Albert, R. (2016). Node-independent elementary signaling modes: a measure of redundancy in Boolean signaling transduction networks. *Netw. Sci.* 4, 273–292. doi: 10.1017/nws.2016.4

Thomas, R., and D'Ari, R. (1990). *Biological Feedback*. Boca Raton, FL: CRC Press.

Tyson, J. J., Baumann, W. T., Chen, C., Verdugo, A., Tavassoly, I., Wang, Y., Weiner, L. M., and Clarke, R. (2011). Dynamic modelling of oestrogen signalling and cell fate in breast cancer cells. *Nat. Rev. Cancer* 11:523–532. doi: 10.1038/nrc3081

Tyson, J. J., Chen, K. C., and Novak, B. (2003). Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell. *Curr. Opin. Cell Biol.* 15, 221–231. doi: 10.1016/S0955-0674(03)00017-6

Wang, L.-Z., Su, R.-Q., Huang, Z.-G., Wang, X., Wang, W.-X., Grebogi, C., et al. (2016). A geometrical approach to control and controllability of nonlinear dynamical networks. *Nat. Commun.* 7:11323. doi: 10.1038/ncomms11323

Wang, R.-S., and Albert, R. (2011). Elementary signaling modes predict the essentiality of signal transduction network components. *BMC Syst. Biol.* 5:44. doi: 10.1186/1752-0509-5-44

Wang, R.-S., Saadatpour, A., and Albert, R. (2012). Boolean modeling in systems biology: an overview of methodology and applications. *Phys. Biol.* 9:055001. doi: 10.1088/1478-3975/9/5/055001

Wang, R.-S., Sun, Z., and Albert, R. (2013). Minimal functional routes in directed graphs with dependent edges. *Int. Trans. Oper. Res.* 20, 391–409. doi: 10.1111/itor.12007

Wells, D. K., Kath, W. L., and Motter, A. E. (2015). Control of stochastic and induced switching in biophysical networks. *Phys. Rev. X* 5:031036. doi: 10.1103/PhysRevX.5.031036

Wynn, M. L., Consul, N., Merajver, S. D., and Schnell, S. (2012). Logic-based models in systems biology: a predictive and parameter-free network analysis method. *Integr. Biol.* 4, 1323–1337. doi: 10.1039/c2ib20193c

Yang, G., Campbell, C., and Albert, R. (2016). Compensatory interactions to stabilize multiple steady states or mitigate the effects of multiple deregulations in biological networks. *Phys. Rev. E* 94:062316. doi: 10.1103/PhysRevE.94.062316

Zañudo, J. G. T., and Albert, R. (2013). An effective network reduction approach to find the dynamical repertoire of discrete dynamic networks. *Chaos* 23:025111. doi: 10.1063/1.4809777

Zañudo, J. G. T., and Albert, R. (2015). Cell fate reprogramming by control of intracellular network dynamics. *PLoS Comput. Biol.* 11:e1004193. doi: 10.1371/journal.pcbi.1004193

Zañudo, J. G., Steinway, S. N., and Albert, R. (2018). Discrete dynamic network modeling of oncogenic signaling: mechanistic insights for personalized treatment of cancer. *Current Opinion in Systems Biology*, 9, 1–10.

Zañudo, J. G. T., Yang, G., and Albert, R. (2017). Structure-based control of complex networks with nonlinear dynamics. *Proc. Natl. Acad. Sci. U.S.A.* 114, 7234–7239. doi: 10.1073/pnas.1617387114

Zertuche, F. (2009). On the robustness of NK-kauffman networks against changes in their connections and boolean functions. *J. Math. Phys.* 50:043513. doi: 10.1063/1.3116166

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

ORIGINAL RESEARCH
published: 22 May 2018
doi: 10.3389/fphys.2018.00550

Using Regularization to Infer Cell Line Specificity in Logical Network Models of Signaling Pathways

*Sébastien De Landtsheer, Philippe Lucarelli and Thomas Sauter**

Systems Biology Group, Life Sciences Research Unit, University of Luxembourg, Belvaux, Luxembourg

Understanding the functional properties of cells of different origins is a long-standing challenge of personalized medicine. Especially in cancer, the high heterogeneity observed in patients slows down the development of effective cures. The molecular differences between cell types or between healthy and diseased cellular states are usually determined by the wiring of regulatory networks. Understanding these molecular and cellular differences at the systems level would improve patient stratification and facilitate the design of rational intervention strategies. Models of cellular regulatory networks frequently make weak assumptions about the distribution of model parameters across cell types or patients. These assumptions are usually expressed in the form of regularization of the objective function of the optimization problem. We propose a new method of regularization for network models of signaling pathways based on the local density of the inferred parameter values within the parameter space. Our method reduces the complexity of models by creating groups of cell line-specific parameters which can then be optimized together. We demonstrate the use of our method by recovering the correct topology and inferring accurate values of the parameters of a small synthetic model. To show the value of our method in a realistic setting, we re-analyze a recently published phosphoproteomic dataset from a panel of 14 colon cancer cell lines. We conclude that our method efficiently reduces model complexity and helps recovering context-specific regulatory information.

Keywords: regularization, sparsity, clustering, network model, logical model, optimization

OPEN ACCESS

Edited by:
Matteo Barberis,
University of Amsterdam, Netherlands

Reviewed by:
Hans A. Kestler,
Universität Ulm, Germany
Alexey Goltsov,
Abertay University, United Kingdom

***Correspondence:**
Thomas Sauter
thomas.sauter@uni.lu

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: *23 February 2018*
Accepted: *30 April 2018*
Published: *22 May 2018*

Citation:
De Landtsheer S, Lucarelli P and Sauter T (2018) Using Regularization to Infer Cell Line Specificity in Logical Network Models of Signaling Pathways. Front. Physiol. 9:550. doi: 10.3389/fphys.2018.00550

1. INTRODUCTION

One goal of Systems Biology is to understand emerging functional properties of biological systems from the interactions of their components (Wolkenhauer, 2014). Such understanding would allow the design of new pharmacological strategies to treat diseases that arise when these systems do not function adequately, like cancer. One frequent approach is to map experimental measurements to the model variables of the system, and infer the most likely parametrization. To be useful, a well-parametrized model of a complex system should not only be able to predict non-obvious, non-linear behaviors, but also provide a mechanistic explanation for these behaviors and to suggest hypotheses about ways to control the system.

The most informative modeling approaches include prior information about the system (Aldridge et al., 2006). Classically, dynamical systems like regulatory networks of mammalian cells are modeled with systems of ordinary differential equations, describing in detail the status

of chemical species like proteins or membrane receptors over time. Alternatively, logical models (Morris et al., 2010; Hill et al., 2012; Le Novère, 2015) were introduced several decades ago for the modeling of regulatory networks (Kauffman, 1969). As they are simpler in their formulation, they are easier to handle computationally, scale better to large models and datasets, and are easier to interpret. The prior knowledge used to construct logical network models frequently comes from reviewing the literature of a certain mechanism, disease or signaling pathway, and may be summarized in a database like STRING, Reactome or WikiPathways (Joshi-Tope et al., 2005; Kutmon et al., 2016; Rigden et al., 2016; Szklarczyk et al., 2017).

Logical models can be used to model stochastic processes. Probabilistic Boolean Networks (Shmulevich et al., 2002) have been introduced to simulate logical models in the presence of uncertainty, as they allow combining multiple Boolean networks with the respective continuous selection probabilities in one mathematical model. They have successfully been applied to the modeling of biological regulatory networks (Trairatphisan et al., 2013). This framework can be generalized to Dynamic Bayesian Networks (DBNs), a general class of models that includes Hidden Markov models and Kalman filters (Murphy, 2002), and can be used to represent the same joint probabilities between variables. In a graphical model of a DBN, the values of the different nodes represent the probabilities for randomly chosen molecules to be in an active state, while the edges represent the probabilities of the parent nodes to activate their targets. Network update is performed according to the laws of probabilities.

There is, however, a number of impediments to successful biomolecular modeling. Firstly, the prior knowledge used to build the model could be inaccurate, or more frequently, incomplete, or both. In other words, compared to the true network, databases likely contain additional edges, as well as miss others. Secondly, the information contained in databases is often generic, collected across cell types, genetic backgrounds, and experimental conditions. Given an interaction graph and a series of contexts (cell types, patients), the task of determining which interactions are context-specific and which ones are context-independent rapidly becomes intractable. This task is however essential to reduce the model complexity, as overly complex models are prone to overfitting (thus less generalizable), computationally expensive, and might be less interpretable than simpler ones. In addition, identification of the most variable model parameters between contexts has the potential to be directly informative about the mechanisms at play and help draw parallels between contexts.

Inter-patient variability is an important factor for many diseases, and in particular cancer. Intra-tumor heterogeneity has been recognized for a long time (Fidler et al., 1982) and it has been established that the heterogeneity of cell lines isolated from different patients spans the genomic, epigenetic, transcriptomic, and proteomic levels, resulting in large phenotypic differences, even within the same tissue of origin (Hoadley et al., 2014). Additionally, the patients' own genetic backgrounds and the tumor micro-environment also play a role in increasing the heterogeneity of clinical responses (Zhou et al., 2008; Marusyk and Polyak, 2011; Junttila and De Sauvage, 2013). However, recent successes in matching a biomarker with the sensitivity to certain targeted anti-cancer therapies, notably in the case of HER2-overexpressing breast cancer (Vogel et al., 2002), EGFR-mutated non-small-cell lung cancer (Lynch et al., 2004), BCR-ABL fusions in chronic myelogeneous leukemia (Sherbenou and Druker, 2007), and $BRAF^{V600E}$-mutant melanoma (Bollag et al., 2010) suggest that the general approach of targeting specific mechanisms in subsets of patients harboring functionally similar tumors is clinically promising.

A number of methods have been devised for the general task of variable selection. Various methods rely on the intuitive notion of comparing models comprising different subsets of the independent variables (Hocking, 1976). This strategy is however problematic for several reasons. Firstly, the number of possible subsets grows very fast with the number of variables, leading to the infeasibility of testing them all. Secondly, repeatedly optimizing a model structure using the same dataset violates the central assumptions of the F-tests or χ^2-based statistics used for comparisons, which are designed to test a single hypothesis. Strategies like forward-selection, backwards elimination, or combinations of both are consequently affected by numerous problems, notably biased parameter estimation and artificially low p-values (Harrell, 2001; Burnham and Anderson, 2002).

Fitting an overspecified model first and clustering the parameters in a second step is not a sound method to achieve sparsity, as the parameter estimates might not be stable, resulting in inaccurate clustering. Furthermore, the two objectives are not coupled, which is problematic: a small difference between the values of two parameters might or might not be supported by the data. It makes more sense to specify our assumptions about the distribution of the parameter values as part of the objective function. Regularization is a technique for adding prior information to a regression problem. It consists in adding to the loss function a function of the parameters alone. More formally, when attempting to learn the parameter set θ from dataset $X = [x_1, x_2, ..., x_n]$ with a model M, the objective function O takes the form:

$$O = f(M(X, \theta), X) + \lambda g(\theta) \tag{1}$$

where f is the loss function, for example the sum of squared errors. The hyperparameter λ is used to balance goodness-of-fit with the regularization objective $g(\theta)$. The most common form of regularization is the Tikhonov regularization (Tikhonov, 1963), also called *ridge regression*, which materializes the assumption that small model parameters are more probable than larger ones. Also called the L_2 norm, the Tikhonov regularization term takes the form:

$$g(\theta) = \sum_{j=1}^{T} (\theta_j)^2 \tag{2}$$

where T is the number of parameters of the model. The L_2 norm is used to impose a penalty on large parameter values. Its popularity is due to the fact that the function is convex, continuous and differentiable everywhere, and is therefore well adapted to gradient descent optimization. It is mostly used in

predictive models to avoid overfitting and produces models that are more generalizable. Because the gradient of this function becomes very small around zero, Tikhonov regularization does not achieve sparsity under most conditions and therefore does not perform variable selection, however this can be solved by the use of thresholds.

Intuitively, the most sensible sparsity constraint should be the L_0 norm, or the cardinality of the non-zero parameter set:

$$g(\theta) = \sum_{j=1}^{T} 1_{(\theta_j \neq 0)} \tag{3}$$

where $1_{(C)}$ is the *indicator* function, and is equal to the number of cases where condition C is true. However, this is usually not feasible in practice, as this function is discontinuous and cannot be used in many optimization algorithms. A good approximation is the L_1 norm, which sums the absolute values of the parameters, without squaring them:

$$g(\theta) = \sum_{j=1}^{T} |\theta_j| \tag{4}$$

The L_1 norm, or LASSO (Tibshirani, 1996) can be used to reduce the size of a model by efficiently removing variables (i.e., set their coefficients to zero) which contribute the least to the model. Importantly, by screening a range of regularization parameter λ, it is possible to order the variables according to their importance. It is natural to use it then, for contextualizing models of biological systems with measurements from different contexts to point to their differences. Different approaches have used the L_1 norm to contextualize network models of signal transduction in mammalian cells. However the assumption is either that there is no relationship between the different cell lines (Eduati et al., 2017; Lucarelli et al., 2018), or that the differences to the mean value should be minimized (Merkle et al., 2016). While the latter works in the case of only two cell lines, it does not when comparing more. The reason is that heterogeneity between cell lines is expected, and we know that different mechanisms are at play in a given experiment. By penalizing any difference, such regularization does not allow parameters to have two or more possible values. However, cancer-related perturbations to molecular interactions occur in discrete steps. Driver mutations often result in the complete loss of the function of a certain protein, for example p53, or constitutive enzymatic activity, for example the common mutation of genes of the RAS family (Kandoth et al., 2013). The desired regularization should therefore penalize differences between contexts but allow for a structure in the parameter space. While a number of methodologies exist (Dondelinger et al., 2012; Hill et al., 2012) to regularize network models of signaling pathways for time-stamped data, in that case the structure of the prior on the parameter space is known, as time is oriented. We propose that the correct assumption for analyzing perturbation data from multiple cell lines, cell types, or across patients, is that network parameter values would form *clusters* corresponding to the most common signaling deregulations. However, methods to efficiently identify the parameters of a biological model and cluster them at the same time are missing.

The general problem of regularizing a model toward a specific, although unknown, structure has been investigated before. The vast majority of the proposed methods combine L_1 and L_2 norms in various ways. Group LASSO (Yuan and Lin, 2006) was introduced to allow the selection of entire groups of variables. This was then extended to a hierarchical selection of nested groups of variables (Zhao et al., 2009), partially overlapping groups of variables (Jacob et al., 2009), and to the induction of sparsity within groups by penalizing for pairwise differences between coefficients of variables belonging to the same group, with the OSCAR algorithm (Bondell and Reich, 2008) and the clustered LASSO (She, 2010). Later Simon et al. proposed the sparse group LASSO (Simon et al., 2012), a modification of the *elastic net* criterion proposed by Zou et al. which combines the L_1 and L_2 norms (Zou and Hastie, 2005). The fused LASSO (Tibshirani et al., 2005) is applicable when there is a natural ordering in the model variables, like time-stamped or spatially organized data. Several groups have tried to decouple the steps of clustering and model fitting, either by considering all possible clusters (Jenatton et al., 2011) or by applying first hierarchical clustering based on the measurements covariance matrix (Bühlmann et al., 2013).

While these approaches have proven useful in some cases (Zhang et al., 2014; Steiert et al., 2016), they do not apply well to the case of regulation networks, because group zero-sparsity (removal of entire groups of variables, as opposed to within-group sparsity) is not necessarily desired, except in the case of network pruning. We therefore implemented a regularized version of the objective function of the FALCON toolbox (De Landtsheer et al., 2017), to lower the degrees of freedom of the model by encouraging the grouping of model parameters across contexts, regardless of the number of groups. This can be achieved by detecting anomalies in the parameter values distribution, assigning a penalty to groups of values more alike the reference null distribution. In our case (Bayesian Networks), the uniform distribution $[0 - 1]$ is assumed to better represent the prior of uncorrelated parameter values, as they are usually interpreted as probabilities. Under different modeling formalisms, other distributions would be more appropriate, for example for ODE-based or constraint-based models. We show how the penality correlates with other measures, with unsupervised clustering, and we demonstrate the use of regularized fitting, first on a small synthetic network model, then with biological data.

2. METHODS

2.1. Algorithm

We propose a measure of uniformity of the parameter values distribution modified from previous work in the field of quasi-random sequences (Sobol, 1976). Given a parameter space $\mathbf{P}$ and N parameter vectors with T parameters $\theta_1, \theta_2, ..., \theta_N$, with $\theta_n = \{\theta_n^1, \theta_n^2, ..., \theta_n^T\}$, we compute for each $t \in T$ the average absolute

deviation from the expected local density of points D_t with:

$$D_t = \sum_{\mathbf{R}\in\mathbf{P}} |1_{(\theta_n^t \in \mathbf{R})} - Vol(\mathbf{R})| \quad (5)$$

for all rectangles $\mathbf{R} = [a_1, b_1] \times [a_2, b_2] \times ... \times [a_T, b_T]$ such that $0 \leq a_i \leq b_i \leq 1$, and with $Vol(\mathbf{R})$ being the volume of the T-dimensional rectangle $\mathbf{R}$.

$$Vol(\mathbf{R}) = \prod_i b_i - a_i \quad (6)$$

The first term in Equation 5 represents the *observed* density of points, while the second one represents the *expected* density. These two quantities are equal in the case of perfect uniformity. We then define the *uniformity* U of the parameter vector as the inverse of the average deviation over the T parameters:

$$U_t = \frac{T}{D_t} \quad (7)$$

and the *uniformity* of an entire model parameter set as the average over all vectors:

$$U = \frac{1}{N}\sum_{i=1}^{N} U_i \quad (8)$$

In one dimension, this metric has an intuitive interpretation, as shown in **Figure 1**: when parameter values are as different as they could be, the expected difference between any two values can be calculated from their relative rank in the set. For example, the distance between two successive observations is $\theta_n^t - \theta_{n-1}^t = 1/N$. When values cluster together, they create windows in which the local density is either higher or lower than this expected value. Note that in one dimension, the rectangles $\mathbf{R}$ are equivalent to the distance between the points, and to the convex hull of these points, while it is not true in higher dimensions.

2.2. Uniformity as a Penality in Regularized Fitting

We analyze the sensitivity of our new metric to the amount of structure in sets of model parameter values by computing it for a large number of sets of uniformly, independently distributed random values. We compare uniformity with the standard deviation, with the results of the Kolmogonov-Smirnov (K-S) (Massey, 1951) and Anderson-Darling (A-D) tests (Anderson and Darling, 1954), and with the sum of pairwise distances. The two non-parametric statistical tests aim at comparing the empirical distribution of the values in the set with a reference distribution, in this case the uniform distribution. The sum of pairwise distances is used in Bondell and Reich (2008) and She (2010), the standard deviation is examplative of measures of spread around a single value, like in Merkle et al. (2016). In addition, we compute for each set the optimal number of clusters (explaining 90% of the variance) using the k-means algorithm and the elbow method (Ketchen and Shook, 1996). Using the inferred number of clusters, we compute the sum of intra-cluster distances. We performed this comparison with 10^4 vectors. Also, to assess the usability of this metric for large-scale computations, we compare the running time of the different computations for sets of size 10, 20, and 40, simulating models with increasing number of contexts.

To illustrate that the use of uniformity as a penalization in an objective function results in the convergence of parameter values into clusters, we iterate a gradient descent process for random sets of uniformly, independently distributed random values. This is equivalent as optimizing a null model using uniformity as a regularizing function, and shows the effect of this penalization in the absence of data. We used gradient descent (using empirical gradients and the interior-point method) with a learning rate of 10^{-3}, collect the shape of the set over 100 updates, and we compare with the centroids of the k-means clustering. All computations were done using Matlab 2017a on a standard desktop computer which specifications are detailed in section 2.3.3.

2.3. Modeling Experiments

Modeling experiments in this paper used the toolbox FALCON (De Landtsheer et al., 2017), a Matlab-based versatile tool to contextualize logical models of regulatory networks. Briefly, FALCON uses a Dynamic Bayesian framework (Lähdesmäki et al., 2006) in which Boolean operations are explicitly defined as arithmetic, continuous logical functions. FALCON emulates a Probabilistic Boolean Network with user-defined topology and uses experimental data from perturbation assays to optimize the weights of the network, which represent the relative activating and inhibiting influences of the network components with respect to the logical functions. For the large-scale analysis of biological data, we used a custom implementation of FALCON running on a high-performance computing platform which specifications are detailed in section 2.3.3.

$$O = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 + \lambda U(\theta) \quad (9)$$

where Y is the vector of measurements for the observed nodes, $\hat{Y}$ is the vector of corresponding predictions and $U(\theta)$ is the uniformity of the parameter set θ across contexts, as defined by Equations 5–8 above, with λ being a scalar that controls the relative contribution of the penality to the objective function. The code and data files used for both the synthetic model and the biological example are available at the address https://github.com/sysbiolux/FALCON. Additional driver scripts are provided in the Supplementary Materials.

2.3.1. Synthetic Toy Model

In order to assess the use of our regularization scheme for finding context-specific parameters, we design a simple toy model with 7 nodes and 9 edges. Two of these nodes are inputs, while two others are measured. We set the model parameters differently for four conceptual cell lines, in such a way that while most parameters are conserved, some would be different, and shared across several (but not all) cell lines. **Figure 2** shows a graphical representation of the network, the values chosen for the model parameters, and the final synthetic data used for model fitting.

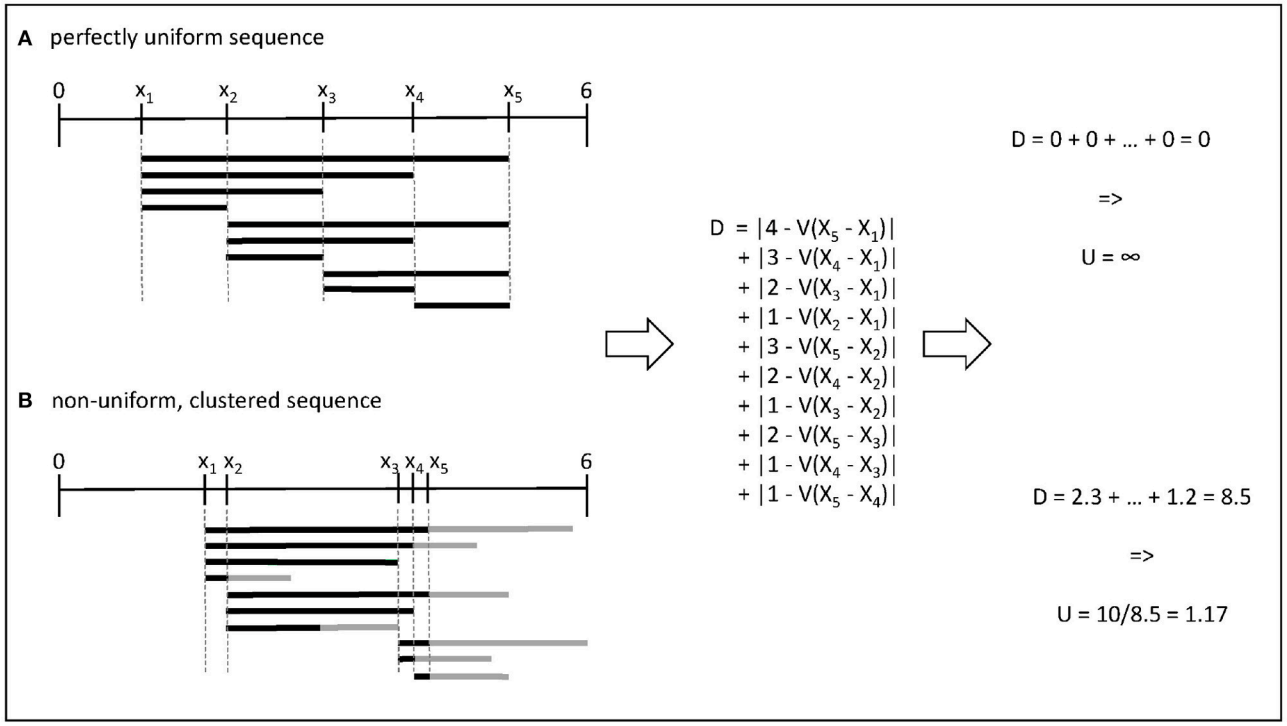
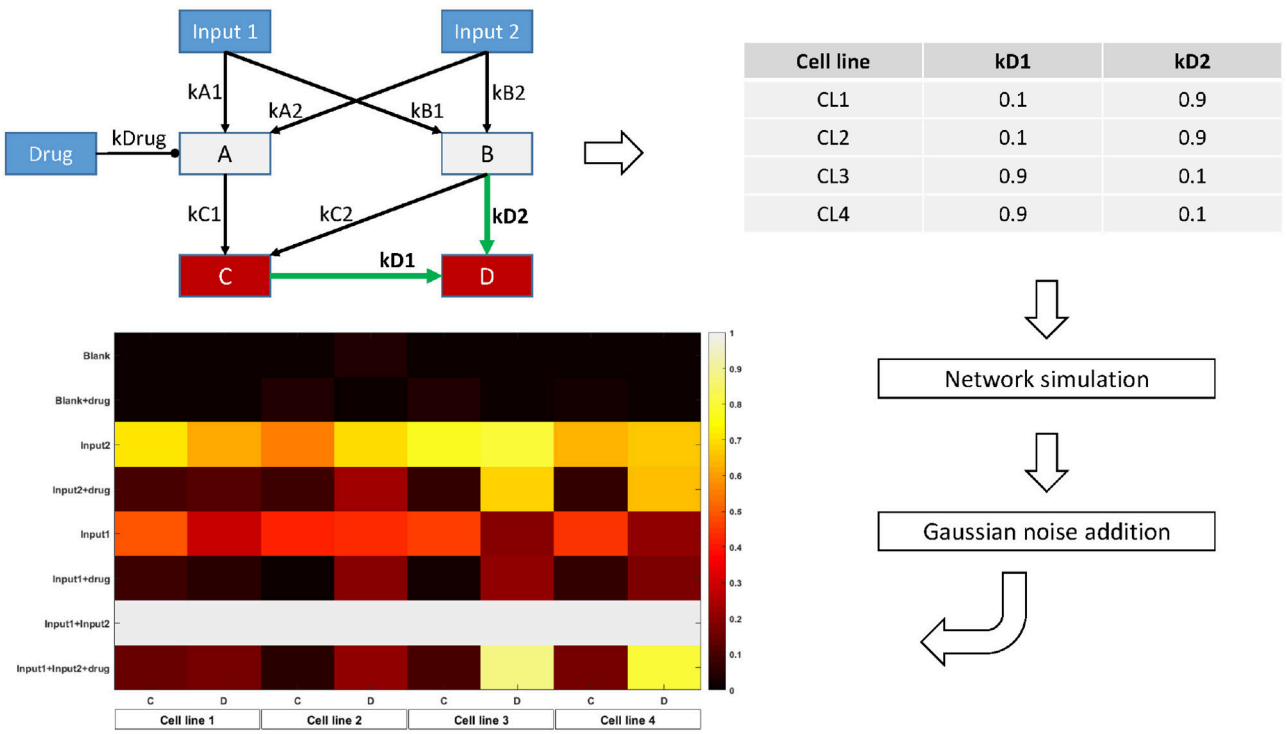


FIGURE 1 | Illustration of the computation of uniformity for two sets of 5 parameter values within the range [0, 6]. **(A)** In the first case, all pairwise distances are equal to the expectation given the rank of the value in the set. **(B)** In the second case, the gray bars indicate the differences compared to the expected density in a given interval.



Cell line	kD1	kD2
CL1	0.1	0.9
CL2	0.1	0.9
CL3	0.9	0.1
CL4	0.9	0.1

FIGURE 2 | Overview of the toy model design. The topology is parametrized in order to display two-by-two similarity between cell lines. For each cell line, the Bayesian Network is simulated with the corresponding parameter values for 8 different combinations of the input nodes values. Random Gaussian noise is added to the values of the two output nodes C and D, simulating biological measurements. The heatmap shows the final node values for each condition, cell line, and node.

To realistically simulate biological data, we use our toy model to generate synthetic steady-state data for the measured nodes by simulating the network with different combinations of values for the input nodes, thereby mimicking a designed, perturbation experiment. We simulate noise in the data by adding a two-component gaussian perturbation around the theoretical value, as explained in Supplementary Methods. The magnitude of the perturbation was chosen to reflect the signal-to-noise ratio of typical biological measurements, for example phosphoproteomics or microarray data.

2.3.2. Biological Dataset

To show the usefulness of our approach in a biological setting, we reanalyze the dataset from Eduati et al. (2017), in which the authors measured 14 phosphoproteins under 43 different perturbed conditions (combinations of 5 stimuli and 7 inhibitors) in 14 colorectal cancer cell lines. Using CellNetOpt (Terfve et al., 2012), they contextualized independent logical ODE models (Wittmann et al., 2009) for each cell line, and proceed to train a statistical model using the cell-specific parameters to predict the responsiveness of the cell lines to a panel of drugs. This study provides an example of the use of system-level analyses to gain understanding of functional properties that cannot be inferred by genomic features alone. We normalized the data (log_2 difference compared to control) linearly to the $[0-1]$ range across cell lines.

Logical ODE models like the one used by Eduati et al. rely on a transformation of the discrete state-space of Boolean models into a continuous one, in such a way that Boolean behavior is preserved on the vertices of the unit cube, i.e., when the inputs are in $\{0, 1\}$. While there are many such possible transformations (Wittmann et al., 2009), the authors chose to use normalized Hill cubes, which are sigmoidal functions of the inputs. The strength of such an approach is the ability to take into account the non-linear 'switch-like' nature of molecular interactions, however at the expense of doubling the number of free parameters (Hill functions are defined by a threshold and a slope). In contrast, our approach uses maximum one parameter per interaction and is restricted to linear relationships, which ensures coherence with the laws of probabilities. To infer the DBN model corresponding to the logical ODE model proposed by Eduati et al., we kept the original topological information, but defined the update function for each node by a multivariate linear function of its parent nodes. In our framework, if two nodes A and B are both activators of a third node X, we have for each time-step t: $X_t = k_A A_{t-1} + k_B B_{t-1}$ with probabilities $0 \leq k_A \leq 1$ and $k_B = 1 - k_A$. Similarly, if a node X is activated by node A but inhibited by node B, we have $X_t = A_{t-1} k_B (1 - B_{t-1})$ with probability $0 \leq k_B \leq 1$.

We used the phosphoprotein data to fit the probabilities for each interaction simultaneously for all cell lines. The complete model comprised 363 nodes and 1106 parameters. The objective function included a penality computed from the average uniformity of the parameters across cell lines, according to Equations 5–8. We optimized 49 models, varying the hyperparameter λ from 2^{-20} to 2^5, and we recovered the optimal parametrization for each cell line in the form of regularization paths. We used the value of 0.01 as threshold for deciding if two parameters should be merged into a single one. For each value of the regularization strength λ, we computed the mean squared error (MSE) and the number of different parameters P in the regularized model, and from these calculate the Bayesian Information Criterion (BIC), which we calculate as $Nlog(MSE) + log(N)P$, with N the number of individual points in the dataset. Lower BIC values indicate models with favorable balance between goodness-of-fit and model complexity (Schwarz, 1978; Burnham and Anderson, 2004).

We selected the model with the lowest BIC for further analyses. We grouped cell line-specific parameters together using the above-mentioned threshold, and re-optimized the model using the obtained topology without the regularization term, in order to obtain unbiased parameter estimates. We performed hierarchical clustering with 1000 bootstrap resamplings on the parameter values using WPGMA and euclidian distance.

Furthermore, we investigated whether the recovered parameter values are associated with drug sensitivity. We downloaded the IC50 values for the 14 cell lines and 83 drugs directly targeting either one of the network's nodes or a target used in clinical practice to treat colorectal cancer from the Genomics of Drug Sensitivity in Cancer database (www.cancerrxgene.org). We computed the linear regression models between each drug and each of the 31 parameters which showed high variability between cell lines (CV $\geq$ 10%). The F-statistic was used to compute a p-value for each test, and q-values were computed from these, using the Benjamini Hochberg procedure to control the False Discovery Rate.

2.3.3. Materials

- Hardware
  - Synthetic model: standard desktop computer equipped with an Intel Xeon E3-1241 CPU clocked at 3.50GHz and 16GB of RAM under Windows 7
  - Biological example: high-performance computing platform with 49 nodes running Matlab2017a, each node consisted of one core of a Xeon-L5640 clocked at 2.26GHz with 3GB RAM
- Software
  - Matlab 2017a (Mathworks, Inc.)
  - FALCON toolbox (https://github.com/sysbiolux/FALCON)
  - Optimization Toolbox (http://nl.mathworks.com/products/optimization/)
  - Parallel Computing Toolbox (http://nl.mathworks.com/help/distcomp/)
  - Bioinformatics toolbox (http://nl.mathworks.com/help/bioinfo/) (optional)

3. RESULTS

3.1. Uniformity as a Measure of Structure

We computed the uniformity U, the standard deviation, the sum of pairwise distances, the K-S statistic, the A-D statistic, and the optimal number of clusters using the k-means algorithm and the elbow method, for 10^4 one-dimensional sets of uniformly, independently distributed random values.

The complete correlation plots are presented in Supplementary Materials. We always show uniformity U on the logarithmic scale. **Figure 3A** shows the relation between uniformity and the standard deviation, while **Figure 3B** shows the correlation between uniformity U and the p-value of the K-S test. Similar results were obtained with the A-D test. The relationship between uniformity, the standard deviation, and the K-S p-value are further explored in **Figure 3C**, and the computation times are compared in **Figure 3D**.

Firstly, $log(U)$ is positively correlated with the p-value of the K-S and A-D non-parametric tests evaluating the distance to the reference uniform distribution, showing that low uniformity is indicative of structure. Secondly, the comparison with the standard deviation shows that low-uniformity sets can have drastically different standard deviations, but that the inverse is not true. This is explained by the fact that sets with tightly clustered values will nevertheless be spread around the global average if there is more than one cluster. **Figure 3C** shows a 3D plot of uniformity, standard deviation, and the K-S p-value and illustrates the point that simple measures of spread are not adapted to the regularization of a set of parameter values if the ground truth is that there is more than one cluster. The figure also displays a graphical representation of the 10 values in the set for four chosen sets, to show that low-uniformity sets correspond to clustered values (with low K-S p-values) while low standard deviation is associated with single clusters.

One important argument for choosing a metric in a regularized optimization problem might be its low computational cost. Comparison of the running time for uniformity with other metrics shows that the new metric can be computed very efficiently (**Figure 3D**), several orders of magnitude faster than the non-parametric tests or the clustering algorithm. This low computational cost makes is well adapted to the repetitive computations characteristic of gradient-descent optimizations.

In addition, we performed experiments using gradient descent either with the standard deviation, sum of pairwise distances, or uniformity U as an objective function on sets of randomly, uniformly distributed random values. Using the regularization objective as the objective function, without data or model to produce an error function, helps understanding the effect of regularization when signal is low in the data. The traces in **Figure 4** reveal the strength and direction of the bias applied on each value in the set in the absence of cost function. Penalizing on the standard deviation results in a homogeneous pull toward the average value (**Figure 4A**), which does not accomplish the goal of forming clusters. Using the sum of pairwise distances, in turn (**Figure 4B**), results in grouping of values together, however the clusters themselves are still pulled together. In contrast, the traces in **Figure 4C** show that using uniformity U, the values form a number of groups, but that these groups are more stable. This is due to the fact that the computation of uniformity U measures local density both below and over the expected value,

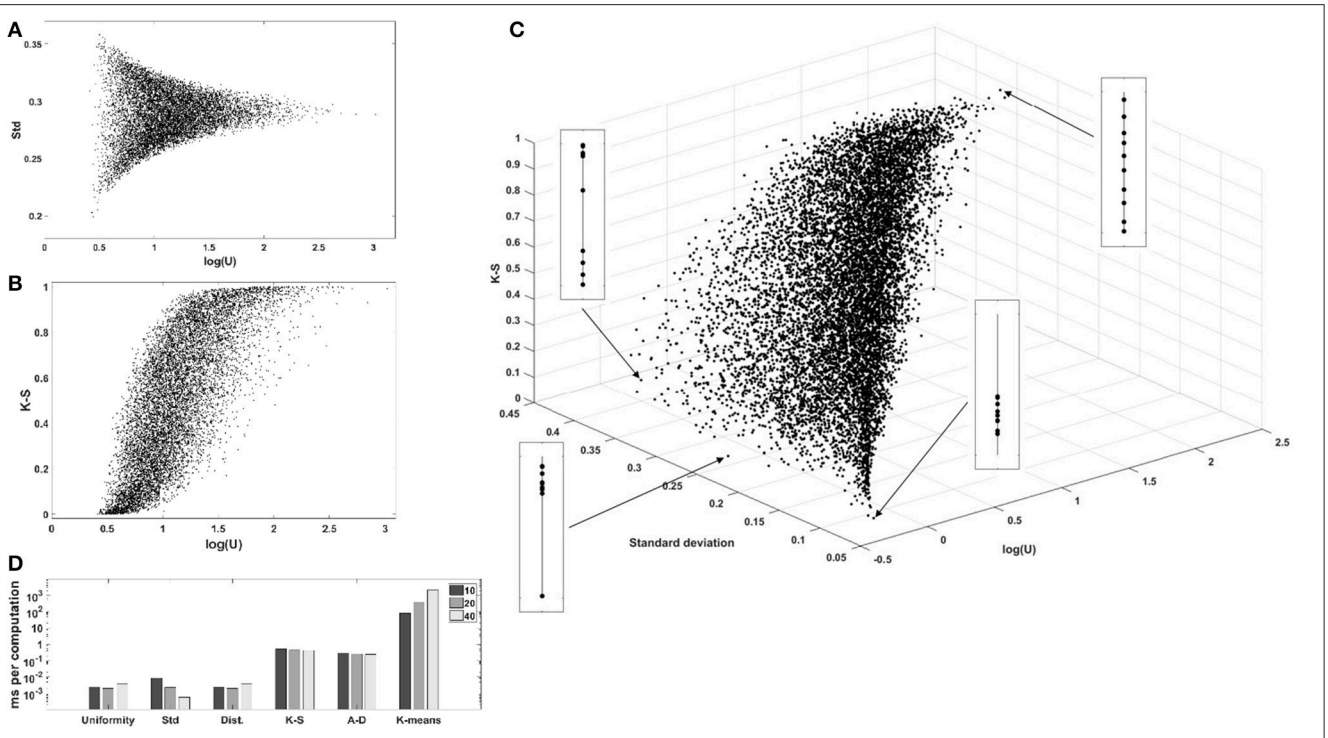


FIGURE 3 | Evaluation of uniformity U as a measure of structure, for 10^4 one-dimensional sets of 10 values. **(A)** Comparison with standard deviation. **(B)** Comparison with the p-value of the K-S test (similar results were obtained with the A-D test). **(C)** 3D-scatterplot of uniformity, standard deviation and K-S p-value. **(D)** Computation times for the different metrics. log(U), $\log_2$(uniformity); Std, standard deviation; Dist, sum of pairwise distances; K-S, p-value of the Kolmogonov-Smirnov test; A-D, p-value of the Anderson-Darling test; K-means, k-means clustering, number of clusters determined with the elbow method.

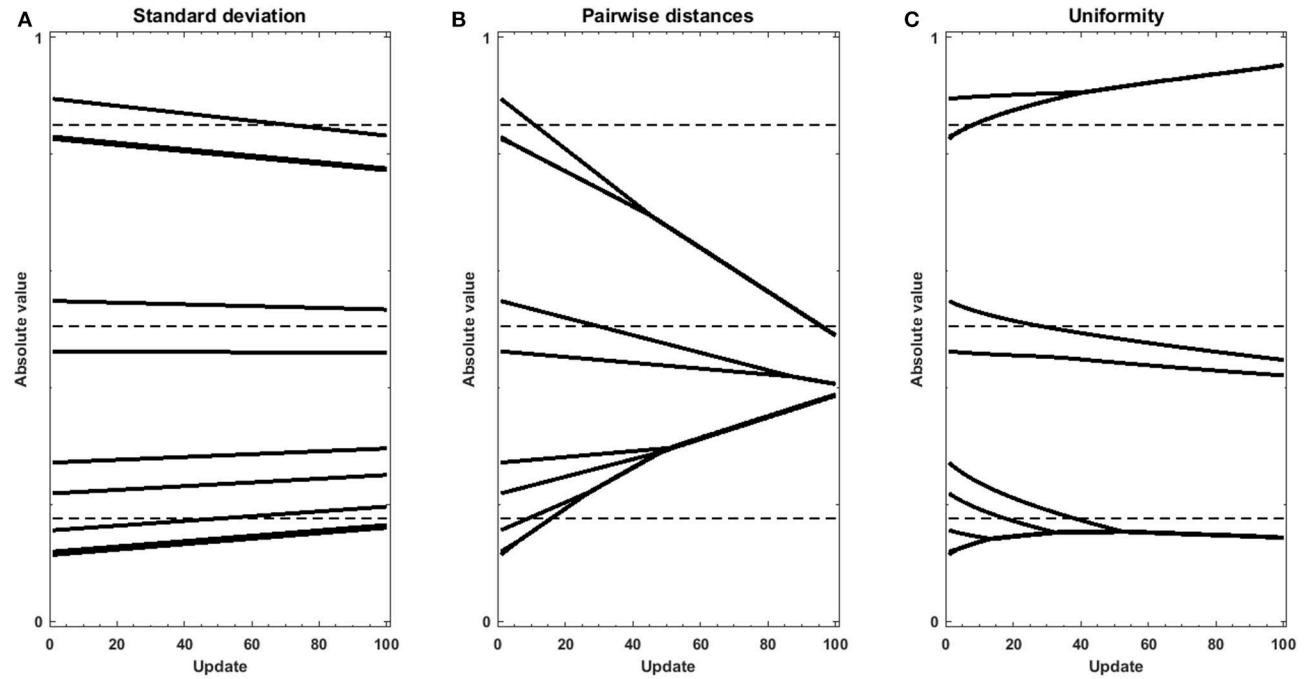


FIGURE 4 | Gradient descent trajectories for a set of randomly uniformly distributed values displaying a certain level of structure, using different metrics as objective function: **(A)** standard deviation, **(B)** sum of pairwise distances and **(C)** Uniformity *U*. The dotted lines show the values of the centroids of clusters as determined by the k-means + elbow method for the original vector.

which means that not only clusters but also voids produce low-uniformity sets. As a result, once values with all clusters have merged, the average of the different clusters remain very similar in number and value to the centroids of the k-means clustering.

3.2. Toy Model

To test the ability of a regularization function using uniformity *U* to recover context-specific parameters of a network model, we generated an example Bayesian Network which we parametrized for four different imaginary contexts. In our example, the contexts are cell lines, and their regulatory network are identically parametrized two by two. We used the network to generate measurements for two of the nodes while two other nodes were controled. We added noise to this synthetic data to simulate background noise and normaly distributed measurement errors. We used the toolbox FALCON to contextualize the network for the four cell lines, with and without regularization based on the uniformity *U* of the set of parameter values. We screened 41 values of the hyperparameter λ. The computations took a total of 220 seconds on a standard desktop computer. The results are presented in **Figure 5**. The regularization paths in **Figure 5A** show the optimal parameter values over a range of regularization strengths λ. The unregularized model is parametrized differently for each cell line, and the regularization induces a grouping of the parameters values across cell lines. However, this clustering occurs at different values of λ. As regularization strength increases, so does the error of the model (**Figure 5B**), while the number of unique parameters in the model decreases as they are merged together. We used the Bayesian Information Criterion to balance goodness-of-fit with model size and identified $\lambda = 2^{-4.5}$ as the best model configuration. **Figures 5D,F** show the fitting cost for each cell line for the unregularized model and the regularized one, respectively. **Figures 5D,G** show the correlation of the simulated values with the measurements, for the unregularized model and the regularized one, respectively, and **Figures 5E,H** show the correlation of the inferred parameter values with the real values for the unregularized model and the regularized one, respectively. Together, these results show that while the new model displays a higher MSE, the inferred parameters are much closer to the ground truth. Regularization transfers a portion of the variance from the parameters back to the data, and so decreases the part of the error on the parameter estimates due to noise. More importantly, the grouping of the samples is easily recovered (Supplementary Figure S2), which also carries information: the cell lines are identical two-by-two.

3.3. Biological Dataset

In order to assess the applicability of our new method of regularization to uncover context-specificity in a realistic modeling setting, we reanalyzed the data from Eduati et al. (2017) using a Dynamic Bayesian Network adapted from the topology of the ODE model. The dataset comprised 8428 datapoints (14 phosphoproteins for 14 cell lines under 43 experimental conditions). We screened 49 values for the hyperparameter λ. The computation time was 1,761 h, or 42 h when parallelized among 49 computing cores. The results are presented in **Figure 6**. Minimum BIC was reached when $\lambda = 0.5$, which corresponds to a model in which 26 of

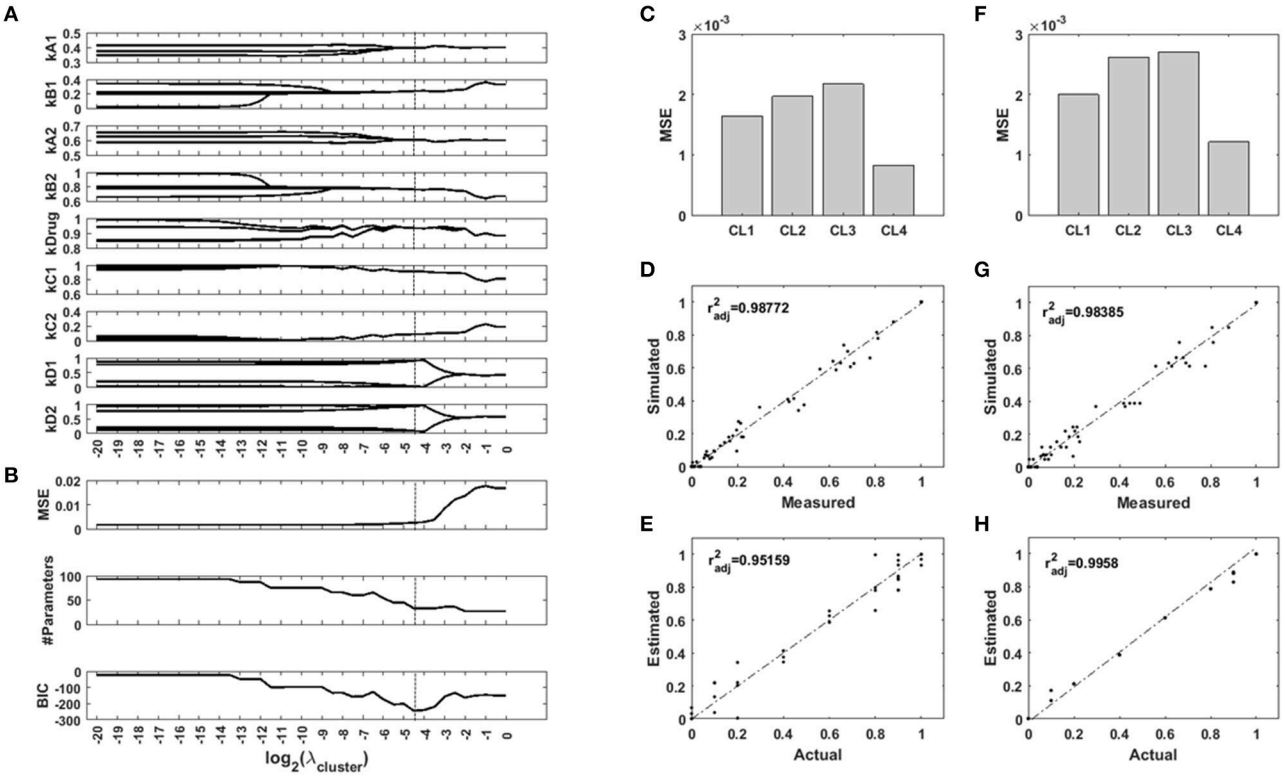


FIGURE 5 | Results of the synthetic toy model analysis. **(A)** Regularization paths for each parameter of the network model. When regularization strength increases, values accross the four contexts are encouraged to merge. **(B)** Mean squared error (MSE), number of different parameters of the model, and Bayesian Information Criterion (BIC) for different regularization strengths. **(C,D,E)** Unregularized model. **(F,G,H)** Sparse model. **(C,F)** MSE for the four contexts with both models. **(D,G)** Comparison of the simulated node values with the measurements for both models. **(E,H)** Comparison of the inferred parameter values with the ground truth for both models. r^2_{adj}: adjusted Pearson's correlation coefficient.

the 79 network parameters can be parametrized identically for all cell lines, and the remaining ones can be organized in 2–9 groups. Overall, the most variable parameter across cell lines is the ERK-EGFR negative feedback (**Figures 6A,B**). Notably, interactions relating to the PI3K/Akt/mTOR axis, to the JUN pathway, and to p38 regulations showed relatively high heterogeneity compared to the crosstalks between them. A number of interactions reveal differential parametrizations for certain cell lines, for example CCK81 in the case of TGFRβ activation by EGFR (**Figure 6C**), or COLO320HSR in the case of RASK activation by IGF1 (**Figure 6D**). **Figure 6E** shows an example of regularization path where no cell line specificity is left in the model with the optimal topology. In addition, many interactions (narrower arrows in **Figure 6A**) show very low values for all cell lines, suggesting that they do not play an important role in this experiment. The complete set of 79 regularization paths is presented in the Supplementary Materials. The changes in BIC are shown in **Figure 6F**, displaying a marked minimum around the value 0.5. The goodness-of-fit was similar for all cell lines, with MSE values ranging from 0.018 to 0.035 (**Figure 6G**). While these results are in line with the ones reported in the original study, it should be noted that in our final model, the role of TAK1 is less prominent, a fact that can be explained by the difference of modeling paradigm. Indeed, while in Eduati et al. (2017) TAK1's *node responsiveness* parameter τ is extremely low for all cell lines while edges from and to TAK1 are quite variable, our modeling framework considers all nodes equally responsive, and as a consequence low TAK1 activity is represented by low edge parameter values.

Figure 6H shows a heatmap of all model parameters for all cell lines. The dendrograms show the clustering of model parameters and cell lines based on their parameter values. We chose WPGMA to perform hierarchical clustering using the euclidian distance between parameter vectors, with 1000 bootstrap replicates. The support for the nodes in the cell line dendrogram are indicated as percentages. Interestingly, cell lines HT29 and HT115 cluster strongly together, while they are highly dissimilar in their genomic alterations. In general, we noted a poor correlation between the genomic and functional pattern over this set of cell lines, a fact already noted in the original study. Cell lines COLO320HSR and CCK81 are the cell lines functionally most unlike the others. This is also visible in the raw data (see Supplementary Materials), notably in the amplitude of the Akt/PI3k/MEK activations.

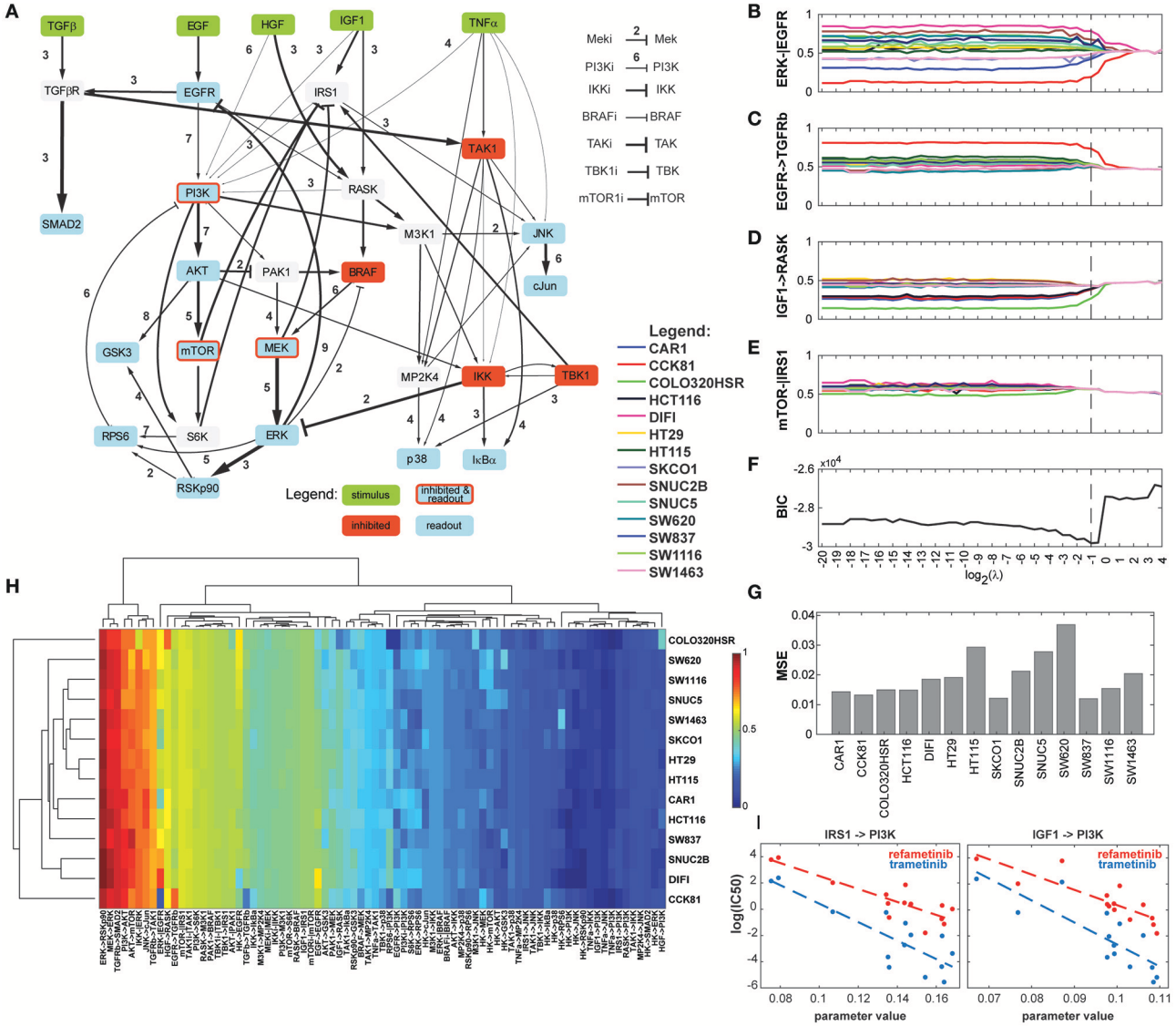


FIGURE 6 | Results of the analysis of the biological dataset. **(A)** Optimized network topology (adapted from Eduati et al., 2017). The width of the arrows represents the median parameter value across the 14 cell lines, with wider arrows corresponding to the most active interactions. The number next to the arrows is the number of clusters that the 14 cell lines form for the optimal regularization strength. **(B–E)** Regularization paths for four chosen interactions, showing decreasing amounts of cell line-specificity. **(F)** BIC (Bayesian Information Criterion) path. **(G)** MSE (Mean Squared Error) for the 14 cell lines for the optimized model. **(H)** Heatmap of the values of the 79 parameters for the 14 cell lines. Dendrograms were produced with WPGMA using euclidian distance. **(I)** Correlation between two PI3K-related parameters and sensitivity to two MEK inhibitors. Left: IRS1-PI3K; refametinib: $r^2 = 0.737$, p-value = 0.133; trametinib: $r^2 = 0.671$, p-value = 0.176; Right: IGF1-PI3K; refametinib: $r^2 = 0.701$, p-value = 0.146; trametinib: $r^2 = 0.652$, p-value = 0.185.

Next, we explored the possible associations between the 31 most variable model parameters and sensitivity to 83 chosen drugs. The 25 most statistically significant of these linear associations are presented in the Supplementary Materials. While no parameter-drug pair shows strong significance (most likely due to the high number of hypotheses tested), we noticed a pattern in which some parameters seem to correlate with sensitivity to MEK inhibitors. **Figure 6I** shows that the parameters relating to PI3K activation by IRS1 and IGF1R are inversely correlated to the log(IC50) of refametinib and trametinib, two known MEK inhibitors.

4. DISCUSSION

We propose a new measure of the degree to which sets of values are clustered around an unknown number of centers. We use this new metric, called uniformity U, as a penalization in the objective function of models of signal transduction. Previously, regularization applied to the parameters of such models have assumed either that parameter values would be mostly identical across the different studied contexts (using measures of spread), and looked for departures from this assumption for context-specific parametrizations, or that the

parameter values would change in correlation with another, known variable between samples (e.g., smoothly over time). While these assumptions make intuitive sense, they are probably not usable in the case of models of regulatory networks in a large number of cell lines. Indeed, functional relationships between molecules in cells, like enzymatic rates and binding strengths, usually exist in a small number of versions for a specific interaction. Because we do not expect these properties to change along a continuum but in a discrete way, it is natural to assume that model parameters of a regulatory network display the same type of structure. Our method efficiently reduces the complexity of network models. In our toy model example, we decrease the number of parameters from 32 to 11, and correctly recover the fact that two groups of cell lines exist and should be parametrized differentially. In our biological example, we decrease the number of parameters from 1,106 to 272, without increasing the error disproportionately.

We show that this method is applicable to biological studies by re-analyzing the dataset from Eduati et al. (2017). Our analysis indicates that the most variable interactions relate to the PI3k/Akt/ERK axis, in particular the ERK/EGFR negative feedback. Interestingly, it has been shown that negative regulation of the MAPK pathway by ERK is a highly complex mechanism and comprises several components, many of which are affected by cancer mutations (Lake et al., 2016).

By performing hierarchical clustering on model parameters after fitting the data to the best model topology, we recover a grouping of the cell lines that correlates poorly with the genomic alterations. We hypothesize that this means we capture a degree of functional heterogeneity that cannot easily be explained by the cell lines' genomic features. Further indication that our network approach is able to recover phenotypical information that is not obvious in the raw measurements is provided by the pattern of relatively strong correlation between a number of parameters and sensitivity to several MEK inhibitors. This observation fits into the recent developments made in integrating network modeling approaches with advanced statistical modeling, where machine-learning methods have been used to successfully predict sensitivity to single drugs and to drug combinations (El-Chaar et al., 2014; Way et al., 2018). Further work is needed to quantify the merits of our regularization scheme when applied in such context.

Our key contribution is the demonstration that using a simple measure of parameter coefficients density inside the parameter space, it is possible to regularize a large network model and to efficiently group together model parameters for which the difference is not well supported by the data. By *de facto* removing part of the noise in parameter estimates, we are able to decrease model complexity. Furthermore, our regularization scheme is easily adaptable to stronger or weaker priors. Equation 8 can be modified as follows:

$$U = \frac{1}{N}\sum_{i=1}^{N} U_i w_i \tag{10}$$

with w being the set of relative weights for the different parameters. When $w_i = 1 \forall i$, all parameters are regularized with the same strength. This weighted average allows the specification of additional prior information, namely that the structural assumptions might not be true everywhere, or that our confidence in these assumptions might be stronger in some cases than others.

It is likely that in the near future, single-cell proteomic studies will provide ever-larger datasets, therefore challenging modeling formalisms and requiring them to adapt to larger number of features (Spitzer and Nolan, 2016). While statistical analyses have largely benefited from regularized parametrizations in the form of more predictive models, the current regularization objectives are not well adapted to the study of signaling networks.

A natural extension of this regularization scheme is to consider subsets of M parameters, corresponding to coherent parts of the model, like known signaling pathways. In that case, regularization will act simultaneously on the different constituent parameters of the pathway, and will allow the determination on cell line-specific pathway activity, a high-level information which is usually recovered by ontology-based pathway analysis. However, in such two-step analysis, the confidence for the different parameters is lost. In addition, ontology-based analyses use pathway knowledge from databases, thus suffer from their incompleteness and inaccuracy.

Finally, although we have demonstrated the applicability of this novel method to the study of regulatory networks with logical models, it would be straightforward to extend its use to other modeling environments. For example, systems of ODEs, which are often used to model regulatory networks, might benefit from the addition of a new kind of regularization, using the same methodology presented in this paper. More generally, regularization based on the uniformity of coefficients would in principle be applicable to any type of regression problem and therefore has the potential to be integrated in many analytical frameworks, and be relevant to advanced statistical analysis.

AUTHOR CONTRIBUTIONS

SD conceived the study, conducted experiments, and wrote the manuscript. PL proposed critical improvements, conducted experiments, and helped editing the manuscript. TS supervised the study, improved the study design, the experimental design and the manuscript.

FUNDING

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 642295 (MEL-PLEX) and the Luxembourg National Research Fund (FNR) within the projects MelanomSensitivity (BMBF/BM/7643621) and ALgoReCEll (INTER/ANR/15/11191283).

ACKNOWLEDGMENTS

The authors would like to acknowledge Dr. Thomas Pfau for technical help with the computations and Dr. Jun Pang for valuable comments on the manuscript.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphys.2018.00550/full#supplementary-material

REFERENCES

Aldridge, B. B., Burke, J. M., Lauffenburger, D. A., and Sorger, P. K. (2006). Physicochemical modelling of cell signalling pathways. *Nat. Cell Biol.* 8, 1195–1203. doi: 10.1038/ncb1497

Anderson, T. W., and Darling, A. D. (1954). A test of goodness of fit. *J. Am. Statist. Assoc.* 49, 765–769. doi: 10.1080/01621459.1954.10501232

Bollag, G., Hirth, P., Tsai, J., Zhang, J., Ibrahim, P. N., Cho, H., et al. (2010). Clinical efficacy of a RAF inhibitor needs broad target blockade in BRAF-mutant melanoma. *Nature* 467, 596–599. doi: 10.1038/nature09454

Bondell, H. D., and Reich, B. J. (2008). Simultaneous regression shrinkage, variable selection and clustering of predictors with OSCAR. *Biometrics* 64, 115–123. doi: 10.1111/j.1541-0420.2007.00843.x

Bühlmann, P., Rütimann, P., van de Geer, S., and Zhang, C. H. (2013). Correlated variables in regression: Clustering and sparse estimation. *J. Statist. Plann. Infer.* 143, 1835–1858. doi: 10.1016/j.jspi.2013.05.019

Burnham, K. P., and Anderson, D. R. (2002). *Model Selection and Multimodel Inference, A Practical Information-Theoretic Approach*. New York, NY: Springer-Verlag.

Burnham, K. P., and Anderson, D. R. (2004). Multimodel inference: understanding AIC and BIC in model selection. *Sociol. Methods Res.* 33, 261–304. doi: 10.1177/0049124104268644

De Landtsheer, S., Trairatphisan, P., Lucarelli, P., and Sauter, T. (2017). FALCON: a toolbox for the fast contextualization of logical networks. *Bioinformatics* 33, 3431–3436. doi: 10.1093/bioinformatics/btx380

Dondelinger, F., Lèbre, S., and Husmeier, D. (2012). Non-homogeneous dynamic Bayesian networks with Bayesian regularization for inferring gene regulatory networks with gradually time-varying structure. *Mach. Learn.* 90, 191–230. doi: 10.1007/s10994-012-5311-x

Eduati, F., Doldàn-Martelli, V., Klinger, B., Cokelaer, T., Sieber, A., Kogera, F., et al. (2017). Drug resistance mechanisms in colorectal cancer dissected with cell type-specific dynamic logic models. *Cancer Res.* 77, 3364–3375. doi: 10.1158/0008-5472.CAN-17-0078

El-Chaar, N. N., Piccolo, S. R., Boucher, K. M., Cohen, A. L., Chang, J. T., Moos, P. J., et al. (2014). Genomic classification of the RAS network identifies a personalized treatment strategy for lung cancer. *Mol. Oncol.* 8, 1339–1354. doi: 10.1016/j.molonc.2014.05.005

Fidler, I. J., Hart, I. R., Fidler, I. J., and Hart, I. R. (1982). Biological diversity in metastatic neoplasms: origins and implications. *Science* 217, 998–1003. doi: 10.1126/science.7112116

Harrell, F. E. (2001). *Regression Modeling Strategies*. Nashville, TN: Springer.

Hill, S. M., Lu, Y., Molina, J., Heiser, L. M., Spellman, P. T., Speed, T. P., et al. (2012). Bayesian inference of signaling network topology in a cancer cell line. *Bioinformatics* 28, 2804–2810. doi: 10.1093/bioinformatics/bts514

Hoadley, K. A., Yau, C., Wolf, D. M., Cherniack, A. D., Tamborero, D., Ng, S., et al. (2014). Multi-platform analysis of 12 cancer types reveals molecular classification within and across tissues-of-origin. *Cell* 158, 929–944. doi: 10.1016/j.cell.2014.06.049

Hocking, R. R. (1976). The analysis and selection of variables in linear regression. *Biometrics* 32, 1–49. doi: 10.2307/2529336

Jacob, L., Obozinski, G., and Vert, J.-P. (2009). "Group lasso with overlap and graph lasso," in *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09* (New York, NY: ACM), 1–8.

Jenatton, R., Audibert, J.-Y., and Bach, F. (2011). Structured variable Selection with Sparsity-Inducing Norms. *J. Mach. Learn. Res.* 12, 2777–2824.

Joshi-Tope, G., Gillespie, M., Vastrik, I., D'Eustachio, P., Schmidt, E., de Bono, B., et al. (2005). Reactome: a knowledgebase of biological pathways. *Nucleic Acids Res.* 33, 428–432. doi: 10.1093/nar/gki072

Junttila, M. R., and De Sauvage, F. J. (2013). Influence of tumour microenvironment heterogeneity on therapeutic response. *Nature* 501, 346–354. doi: 10.1038/nature12626

Kandoth, C., McLellan, M. D., Vandin, F., Ye, K., Niu, B., Lu, C., et al. (2013). Mutational landscape and significance across 12 major cancer types. *Nature* 502, 333–339. doi: 10.1038/nature12634

Kauffman, S. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theoret. Biol.* 22, 437–467. doi: 10.1016/0022-5193(69)90015-0

Ketchen, D. J. J., and Shook, C. L. (1996). The application of cluster analysis. *Strat. Manag. J.* 17, 441–458. doi: 10.1002/(SICI)1097-0266(199606)17:6<441::AID-SMJ819>3.0.CO;2-G

Kutmon, M., Riutta, A., Nunes, N., Hanspers, K., Willighagen, E. L., Bohler, A., et al. (2016). WikiPathways: capturing the full diversity of pathway knowledge. *Nucleic Acids Res.* 44, D488–D494. doi: 10.1093/nar/gkv1024

Lähdesmäki, H., Hautaniemi, S., Shmulevich, I., and Yli-Harja, O. (2006). Relationships between probabilistic Boolean networks and dynamic Bayesian networks as models of gene regulatory networks. *Signal Proces.* 86, 814–834. doi: 10.1016/j.sigpro.2005.06.008

Lake, D., Corrêa, S. A. L., and Müller, J. (2016). Negative feedback regulation of the ERK1/2 MAPK pathway. *Cell. Mol. Life Sci.* 73, 4397–4413. doi: 10.1007/s00018-016-2297-8

Le Novère, N. (2015). Quantitative and logic modelling of molecular and gene networks. *Nat. Rev. Genet.* 16, 146–158. doi: 10.1038/nrg3885

Lucarelli, P., Schilling, M., Kreutz, C., Vlasov, A., Boehm, M. E., Iwamoto, N., et al. (2018). Resolving the combinatorial complexity of smad protein complex formation and its link to gene expression. *Cell Syst.* 6, 75–89. doi: 10.1016/j.cels.2017.11.010

Lynch, T. J., Bell, D. W., Sordella, R., Gurubhagavatula, S., Okimoto, R. A., Brannigan, B. W., et al. (2004). Activating mutations in the epidermal growth factor receptor underlying responsiveness of non-small-cell lung cancer to gefitinib. *New Engl. J. Med.* 350, 2129–2139. doi: 10.1056/NEJMoa040938

Marusyk, A., and Polyak, K. (2011). Tumor heterogeneity: causes and consequences. *Biochim. Biophys. Acta* 1805, 105–117. doi: 10.1016/j.bbcan.2009.11.002

Massey, F. J. (1951). The Kolmogorov-Smirnov Test for Goodness of Fit. *J. Am. Stat. Assoc.* 46, 68–78. doi: 10.1080/01621459.1951.10500769

Merkle, R., Steiert, B., Salopiata, F., Depner, S., Raue, A., Iwamoto, N., et al. (2016). Identification of cell type-specific differences in erythropoietin receptor signaling in primary erythroid and lung cancer cells. *PLoS Comput. Biol.* 12:e1005049. doi: 10.1371/journal.pcbi.1005049

Morris, M. K., Saez-Rodriguez, J., Sorger, P. K., and Lauffenburger, D. A. (2010). Logic-based models for the analysis of cell signaling networks. *Biochemistry* 49, 3216–3224. doi: 10.1021/bi902202q

Murphy, K. P. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, CA.

Rigden, D. J., Fernández-Suárez, X. M., and Galperin, M. Y. (2016). The 2016 database issue of nucleic acids research and an updated molecular biology database collection. *Nucleic Acids Res.* 44, D1–D6. doi: 10.1093/nar/gkv1356

Schwarz, G. (1978). Estimating the dimension of a model. *Annal. Stat.* 6, 461–464. doi: 10.1214/aos/1176344136

She, Y. (2010). Sparse regression with exact clustering. *Electr. J. Stat.* 4, 1055–1096. doi: 10.1214/10-EJS578

Sherbenou, D. W., and Druker, B. J. (2007). Applying the discovery of the Philadelphia chromosome. *J. Clin. Invest.* 117, 2067–2074. doi: 10.1172/JCI31988

Shmulevich, I., Dougherty, E. R., Kim, S., and Zhang, W. (2002). Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics* 18, 261–274. doi: 10.1093/bioinformatics/18.2.261

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2012). A sparse-group lasso. *J. Comput. Graph. Stat.* 22, 231–245. doi: 10.1080/10618600.2012.681250

Sobol, I. M. (1976). Uniformly distributed sequences with an additional uniform property. *USSR Comput. Math. Math. Phys.* 16, 236–242. doi: 10.1016/0041-5553(76)90154-3

Spitzer, M. H., and Nolan, G. P. (2016). Mass cytometry: single cells, many features. *Cell* 165, 780–791. doi: 10.1016/j.cell.2016.04.019

Steiert, B., Timmer, J., and Kreutz, C. (2016). L1 regularization facilitates detection of cell type-specific parameters in dynamical systems. *Bioinformatics* 32, i718–i726. doi: 10.1093/bioinformatics/btw461

Szklarczyk, D., Morris, J. H., Cook, H., Kuhn, M., Wyder, S., Simonovic, M., et al. (2017). The STRING database in 2017: quality-controlled protein-protein association networks, made broadly accessible. *Nucleic Acids Res.* 45, D362–D368. doi: 10.1093/nar/gkw937

Terfve, C., Cokelaer, T., Henriques, D., MacNamara, A., Goncalves, E., Morris, M. K., et al. (2012). CellNOptR: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms. *BMC Syst. Biol.* 6:133. doi: 10.1186/1752-0509-6-133

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *J. R. Stat. Soc. Ser. B* 58, 267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., Knight, K., and Watson, I. B. M. T. J. (2005). Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc. Ser. B* 67, 91–108. doi: 10.1111/j.1467-9868.2005.00490.x

Tikhonov, A. N. (1963). Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.* 4, 1035–1038.

Trairatphisan, P., Mizera, A., Pang, J., Tantar, A. A., Schneider, J., and Sauter, T. (2013). Recent development and biomedical applications of probabilistic Boolean networks. *Cell Commun. Sign.* 11:46. doi: 10.1186/1478-811X-11-46

Vogel, C. L., Cobleigh, M. A., Tripathy, D., Gutheil, J. C., Harris, L. N., Fehrenbacher, L., et al. (2002). Efficacy and Safety of Trastuzumab as a Single Agent in First-Lin Treatment of HER2-Overexpressing Metastatic Breast Cancer. *J. Clin. Oncol.* 20, 719–726. doi: 10.1200/JCO.2002.20.3.719

Way, G. P., Sanchez-Vega, F., La, K., Armenia, J., Chatila, W. K., Luna, A., et al. (2018). Machine learning detects pan-cancer ras pathway activation in the cancer genome atlas. *Cell Reports* 23, 172–180.e3. doi: 10.1016/j.celrep.2018.03.046

Wittmann, D. M., Krumsiek, J., Saez-Rodriguez, J., Lauffenburger, D. A., Klamt, S., and Theis, F. J. (2009). Transforming Boolean models to continuous models: methodology and application to T-cell receptor signaling. *BMC Syst. Biol.* 3:98. doi: 10.1186/1752-0509-3-98

Wolkenhauer, O. (2014). Why model? *Front. Physiol.* 5:21. doi: 10.3389/fphys.2014.00021

Yuan, M., and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B Stat. Methodol.* 68, 49–67. doi: 10.1111/j.1467-9868.2005.00532.x

Zhang, L., Baladandayuthapani, V., Mallick, B. K., Manyam, G. C., Thompson, P. A., Bondy, M. L., et al. (2014). Bayesian hierarchical structured variable selection methods with application to molecular inversion probe studies in breast cancer. *J. R. Stat. Soc. Ser. C Appl. Stat.* 63, 595–620. doi: 10.1111/rssc.12053

Zhao, P., Rocha, G., and Yu, B. (2009). The composite absolute penalties family for grouped and hierarchical variable selection. *Annal. Stat.* 37, 3468–3497. doi: 10.1214/07-AOS584

Zhou, S. F., Di, Y. M., Chan, E., Du, Y. M., Chow, V. D. W., Xue, C. C. L., et al. (2008). Clinical pharmacogenetics and potential application in personalized medicine. *Curr. Drug Metab.* 9, 738–784. doi: 10.2174/138920008786049302

Zou, H., and Hastie, T. (2005). Addendum: regularization and variable selection via the elastic net. *J. R. Statist. Soc. B* 67, 301–320. doi: 10.1111/j.1467-9868.2005.00503.x

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

METHODS
published: 23 May 2018
doi: 10.3389/fphys.2018.00549

DSGRN: Examining the Dynamics of Families of Logical Models

Bree Cummins[1], *Tomas Gedeon*[1]*, *Shaun Harker*[2] *and Konstantin Mischaikow*[2]

[1] Department of Mathematical Sciences, Montana State University, Bozeman, MT, United States, [2] Department of Mathematics, Rutgers, The State University of New Jersey, New Brunswick, NJ, United States

We present a computational tool DSGRN for exploring the dynamics of a network by computing summaries of the dynamics of switching models compatible with the network across all parameters. The network can arise directly from a biological problem, or indirectly as the interaction graph of a Boolean model. This tool computes a finite decomposition of parameter space such that for each region, the state transition graph that describes the coarse dynamical behavior of a network is the same. Each of these parameter regions corresponds to a different logical description of the network dynamics. The comparison of dynamics across parameters with experimental data allows the rejection of parameter regimes or entire networks as viable models for representing the underlying regulatory mechanisms. This in turn allows a search through the space of perturbations of a given network for networks that robustly fit the data. These are the first steps toward discovering a network that optimally matches the observed dynamics by searching through the space of networks.

Keywords: Boolean networks, switching systems, network dynamics, parameter space, database of dynamics

OPEN ACCESS

Edited by:
Matteo Barberis,
University of Amsterdam, Netherlands

Reviewed by:
Tomáš Helikar,
University of Nebraska-Lincoln, United States
Kyle B. Gustafson,
United States Department of the Navy, United States
Marija Cvijovic,
Chalmers University of Technology, Sweden

***Correspondence:**
Tomas Gedeon
gedeon@math.montana.edu

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: *31 January 2018*
Accepted: *30 April 2018*
Published: *23 May 2018*

Citation:
Cummins B, Gedeon T, Harker S and Mischaikow K (2018) DSGRN: Examining the Dynamics of Families of Logical Models. Front. Physiol. 9:549. doi: 10.3389/fphys.2018.00549

1. INTRODUCTION

Experimentally determined pairwise interactions between genes, proteins and signaling molecules are often assembled into pathways and networks. There is a strong desire to understand the dynamics of networks, diversity of their potential stable behavior, as well their response under mutations or targeted pharmacological intervention. Such an ability would allow us to target many diseases, most importantly cancer, with great precision and accuracy, without disturbing other functions of the cell, and without the devastating side effects on healthy cells that are the hallmark of many current drugs.

The current state of modeling gene network dynamics is characterized by a trade-off between the model's ability to quantitatively match the experimental data, and the need for a large number of kinetic parameters to parameterize the model (Karlebach and Shamir, 2008; Heatha and Kavria, 2009; Machado et al., 2011; Goncalves et al., 2013). Properly parameterized ordinary differential equation models can provide a good quantitative match and are easily generalized (Chen et al., 2004; Tyson and Novak, 2013). However, numerical simulation of these models require knowledge of kinetic parameters that are usually not known. The indirect estimate of these parameters by comparing the output of the model to the experimental data suffers from at least three fundamental problems: (i) the correspondence between dynamics and the structure of the network is not one-to-one; (ii) the need to match data corrupted by significant intrinsic and experimental noise to an individual solution of the ODE model; and (iii) the lack of methods to search high dimensional parameter spaces for dynamic signatures observed in the data.

A popular modeling platform is that of Boolean nets, where each protein, ligand or mRNA is assumed to have two states (ON and OFF), and the discrete time evolution of the states is based on logic-like update functions (Glass and Kauffman, 1972, 1973; Thomas, 1973; Thomas et al., 1995; von Dassow et al., 2000; Bernard and Gouze, 2002; de Jong, 2002; de Jong et al., 2004; Belta and Habets, 2006; Chaves et al., 2006; Faure et al., 2006; Albert, 2007; Batt et al., 2007a,b; Bornholt, 2008; Tournier and Chaves, 2009; Machado et al., 2011; Albert et al., 2013; Saadatpour and Reka, 2013). Rather than providing rate parameters, the biological input into model formulation is limited to postulating logical functions, one for each node in the network, which compute the next Boolean state of node i based on Boolean states of the nodes that provide input to node i. These Boolean functions at each node are assembled into a Boolean function that provides the next state of all nodes in the network based on the previous state of the network. Iterations of this function are an approximation of the time evolution of the state of the network.

This attractive class of synchronous Boolean models has several disadvantages. The first class of objections comes from biology: these models cannot represent ubiquitous cellular noise, since states change simultaneously they require unreasonable uniformity of duration of different cellular processes, and the fit to experimental data is typically problematic. A mathematical objection is that discretization of the phase space and the discretization of the set of Boolean functions compatible with a given network does not allow consideration of changing dynamics under graded perturbation. In other words, it is difficult to construct a bifurcation theory in the class of Boolean functions.

In this contribution we study multi-level discrete maps, which are a direct generalization of Boolean maps, that are compatible with an ODE system. We propose that only the asynchronous updates of these discrete maps have biological meaning. The concept of an asynchronous update of a Boolean function has been introduced previously (Pauleve and Richard, 2012). We review and formalize these concepts in the next section. We then study a particular class of ODEs that can be viewed as a continuous parameterization of a family of multi-level discrete maps. Continuous parameterization of a finite number of inherently discrete objects implies that there is a finite decomposition of the parameter space into disjoint domains, each of which supports a multi-level discrete map. Mutual position of these parameter domains is captured in a *parameter graph*, whose nodes represent the domains and edges their adjacency.

We describe a computational approach, called Dynamic Signatures Generated by Regulatory Networks (DSGRN), that computes the parameter graph for a given network and input interaction at each node. In addition, to each node of the parameter graph we associate a *Morse graph* whose nodes are the strongly connected path components of the asynchronous update of the corresponding multi-level discrete map, and edges represent reachability by iterations of this map. We call the resulting collection a *DSGRN Database*.

2. BASIC DEFINITIONS

Definition 2.1. A *regulatory network* $\mathbf{RN} = (V, E)$ is a graph with network nodes $V = \{1, 2, \ldots, N\}$ and signed, directed edges $E \subset V \times V \times \{\rightarrow, \dashv\}$. For $i, j \in V$, we will use the notation $(i, j) \in E$ to denote a directed edge from i to j of either sign, $i \rightarrow j$ to denote an *activation* or positive interaction, and $i \dashv j$ to denote a *repression* or negative interaction.

We define the *targets* of a node i as

$$\mathbf{T}(i) := \{j \mid (i, j) \in E\}$$

and the *sources* of a node i as

$$\mathbf{S}(i) := \{j \mid (j, i) \in E\}$$

For each node i in a regulatory network **RN**, define a set of *integer states* $\mathcal{V}(i) := \{0, 1, \ldots, m_i\}$. Let $\mathcal{V} := \prod_{i=1}^{N} \mathcal{V}(i)$. For state $s \in \mathcal{V}$ let

$$S^i_+ := \{u \in \mathcal{V} \mid u_i > s_i, u_j = s_j \text{ for all } j \neq i\}$$

be the set of states that differ from s only in the i-th coordinate and are strictly greater in the i-th coordinate.

Definition 2.2. We say a (multi-valued) map $f : \mathcal{V} \rightarrow \mathcal{V}$ is *compatible* with a regulatory network **RN** (**RN**-compatible) if and only if the following are satisfied

- $(i, j) \in E$ is a positive edge $i \rightarrow j$ if and only if there exists a state $s \in \mathcal{V}$ and at least one $u \in S^i_+$ such that $f_j(u) > f_j(s)$.
- $(i, j) \in E$ is a negative edge $i \dashv j$ if and only if there exists a state $s \in \mathcal{V}$ and at least one $u \in S^i_+$ such that $f_j(u) < f_j(s)$.

A regulatory network, as introduced in this paper, is also called the *interaction graph* of Boolean function f, as defined in Pauleve and Richard (2012). Our definition above goes in the opposite direction and defines a set of multivalued maps consistent with a fixed regulatory network; we also generalize from Boolean maps to maps with more than two discrete values.

Definition 2.3. A *synchronous Boolean model for a regulatory network **RN*** is an **RN**-compatible map

$$B : \{0, 1\}^N \rightarrow \{0, 1\}^N.$$

Given a synchronous Boolean model B, the regulatory network **RN** such that B is **RN**-compatible, is the *interaction graph of* B.

Definition 2.4. A *synchronous multi-level discrete map for a regulatory network **RN*** is an **RN**-compatible map

$$D : \mathcal{V} \rightarrow \mathcal{V}$$

where $\mathcal{V} = \prod_{i=1}^{N} \{0, 1, \ldots, m_i\}$.

Definition 2.5. A *nearest neighbor multi-valued map for a regulatory network **RN*** is an **RN**-compatible map

$$\mathcal{F} : \mathcal{V} \rightrightarrows \mathcal{V}$$

such that either $s \in \mathcal{F}(s)$ or, if $v \in \mathcal{F}(s)$ and $v \neq s$ then v satisfies the *adjacency condition*:

$$|v_i - s_i| = \begin{cases} 1, & \text{for } i = k \\ 0, & \text{for } i \neq k \end{cases}$$

for exactly one index k. We say that s and v are *adjacent*.

Definition 2.6. We say a nearest neighbor multi-valued map $\mathcal{F}$ is *an asynchronous update* of a multi-level discrete map D if, given

$$t_1 = D(s_1) \qquad \text{where} \qquad t_1 = (t_{1,1}, \ldots, t_{1,N}) \text{ and } s_1 = (s_{1,1}, \ldots, s_{1,N}),$$

we have $s_2 \in \mathcal{F}(s_1)$ in either of the two following conditions:

(a) if $t_1 = s_1$ then $s_2 = s_1$; or
(b) if $t_1 \neq s_1$, then s_2 is adjacent to s_1, and s_2 lies between s_1 and t_1, which means that either
   (a) $s_{1,i} < s_{2,i} \leq t_{1,i}$ or
   (b) $s_{1,i} > s_{2,i} \geq t_{1,i}$.

For a regulatory network $\mathbf{RN} = (V, E)$ consider a system of ODEs in variables x_i for each $i \in V$. We assume that there are finite number of thresholds $\theta_{1,i}, \ldots, \theta_{m_i,i}$ that divide the semi-axis $[0, \infty)$ to $m_i + 1$ intervals I_k. The collection of thresholds $\{\theta_{j,i}\}$ decomposes $[0, \infty)^N$ into a finite number of domains κ, characterized by the property that the projection on i-th variable $\pi_i(\kappa) = I_k$ for a unique $k \in \{0, \ldots, m_i\}$ for every i. We call each κ a *domain*. Let $\mathcal{K}$ be a collection of all domains $\kappa \subset \mathbb{R}^{N+}$ in the non-negative orthant of $\mathbb{R}^N$.

Let $x = (x_1, \ldots, x_N) \in \mathbb{R}^{N+}$ and let

$$G_i : [0, \infty) \to \mathcal{V}(i)$$

be defined by $G_i(x_i) = k$ if and only if $x_i \in I_k$. Let

$$G : [0, \infty)^N \to \mathcal{V}$$

be the vector-valued function with coordinate functions G_i. For a given domain κ, the value $G(x)$ does not depend on $x \in \kappa$. Therefore we can assign the *state* $s := G(x) \in \mathcal{V}, x \in \kappa$ to the domain κ and write $s = g(\kappa)$. Viewed as a map on the set of domains $\mathcal{K}$, g is a bijection

$$g : \mathcal{K} \to \mathcal{V}.$$

Definition 2.7. For a regulatory network $\mathbf{RN} = (V, E)$ consider a system of ODEs in variables x_i for each $i \in V$. We say that such an ODE system is *compatible* with a nearest neighbor multi-valued map $\mathcal{F}$ if solutions $x(t)$ can traverse from domain κ_1 to adjacent domain κ_2 only if $g(\kappa_2) \in \mathcal{F} \circ g(\kappa_1)$.

This definition of compatible ODE system states that the dynamics of an ODE system can be captured, in an coarse sense, by a finite multi-valued map. We now apply these ideas to a specific family of ODE systems.

3. SWITCHING SYSTEMS

Switching systems, also known as Glass systems, were introduced by Glass (Glass and Kauffman, 1972, 1973) in the 1970's and developed subsequently by many authors (Thomas, 1973; Thomas et al., 1995; Edwards, 2001; Bernard and Gouze, 2002; de Jong, 2002; de Jong et al., 2004; Chaves et al., 2006; Tournier and Chaves, 2009; Ironi et al., 2011; Edwards et al., 2015).

Definition 3.1. A *switching system* for a regulatory network **RN**$= (V, E)$ is a system of ordinary differential equations

$$\dot{x}_i = -\gamma_i x_i + M_i \circ \sigma_i(x), i \in V \tag{1}$$

where $\gamma_i > 0$ is a decay rate, M_i is a multi-affine algebraic expression (Belta and Habets, 2006; Batt et al., 2007b; Cummins et al., 2016) , and $\sigma_i = (\sigma_{i,j})$ is a vector of step functions, one for each edge $(j, i) \in E$. When $(j, i) = j \to i$ is an activation, then the step function transitions from a low ($l_{i,j}$) to a high value ($u_{i,j}$), and when $(j, i) = j \dashv i$ is a repression, then $\sigma_{i,j}$ transitions from $u_{i,j}$ to $l_{i,j}$. The transition happens at the threshold $x_j = \theta_{j,i}$:

$$\sigma_{i,j} := \begin{cases} l_{i,j} & \text{if } j \to i \in E \text{ and } x_j < \theta_{i,j} \\ & \text{or } j \dashv i \in E \text{ and } x_j > \theta_{i,j} \\ u_{i,j} & \text{if } j \to i \in E \text{ and } x_j > \theta_{i,j} \\ & \text{or } j \dashv i \in E \text{ and } x_j < \theta_{i,j} \end{cases} \tag{2}$$

We assume $0 < \theta_{i,j}$ and $0 < l_{i,j} < u_{i,j}$ to ensure the model captures the basic biological meaning of concentration, activation, and repression. We further assume $\theta_{i,j} \neq \theta_{k,j}$ for all $j \in V$ whenever $i \neq k$ and so each node j affects its downstream nodes at different thresholds.

It is important to note that to a given **RN** one can associate many switching systems. Indeed, a selection of multi-linear expressions $M_i, i = 1, \ldots, N$ in addition to the structure of the network **RN**, determines the parameterized set of ODEs (1). The function M_i determines how the information from the source nodes $\mathbf{S}(i)$ is combined into the right hand side of (1).

A *parameter* of the switching system is a set of real numbers

$$p = \{\gamma_i \mid i \in V\} \cup \{\theta_{i,j}, l_{i,j}, u_{i,j} \mid (j, i) \in E\}$$

that satisfy these constraints. The set of all parameters p is denoted P.

Definition 3.2. The collection $\Theta_i := \{\theta_{j,i} \mid j \in \mathbf{T}(i)\}$ for each node $i \in V$ is totally ordered, and this order induces a decomposition of phase space $\mathcal{K}$, such that each domain $\kappa \in \mathcal{K}$ is written

$$\kappa = \prod_i [\theta_{j_k,i}, \theta_{j_{k+1},i}]$$

where $\theta_{j_k,i}, \theta_{j_{k+1},i}$ are adjacent. We define the thresholds $\theta_{0,i} := 0$ and $\theta_{\infty,i} := \infty$, so that the intervals below the lowest threshold and above the highest threshold are captured.

Let $m_i = |\mathbf{T}(i)|$ be the number of targets of node $i \in V$, and let $\mathcal{V} = \prod_{i=1}^{N}\{0, 1, \ldots, m_i\}$ as before. The decomposition $\mathcal{K}$ is the same as that in the previous section, and using the total order on Θ_i, we can construct an appropriate bijection $g : \mathcal{K} \to \mathcal{V}$. Using this bijection g, we show in Crawford-Kahrl et al. (2018) that given a switching system at a fixed parameter $p \in P$, there is a unique multi-level discrete map D^p, and an asynchronous update rule of D^p, $\mathcal{F}^p$, such that the switching system is compatible with $\mathcal{F}^p$. We note that the collection $\{D^p\}_{p\in P}$ does not exhaust the entire collection of **RN**-compatible multi-level maps D. However, the induced collection of maps $\{D^p\}_{p\in P}$ decomposes into finite number of classes.

Definition 3.3. Let p be a parameter of a switching system with totally ordered thresholds Θ_i^p. Let D^p be the unique multi-level function associated to the switching system parameterized by p. Let $O_i^p = \{j_1 < j_2 < \cdots < j_{m_i}\}$ be such that $j_k < j_l$ if and only if $\theta_{j_k,i} < \theta_{j_l,i}$ in Θ_i^p. Define $O^p = \{O_i^p\}$ to be the *order parameter* associated to p, and (O^p, D^p) to be the *combinatorial parameter* of the system. If q is another parameter of the switching system with (O^q, D^q), then we define an equivalence relation $q \sim p$ when $(O^q, D^q) = (O^p, D^p)$. We call the collection of combinatorial parameters $\mathcal{Z}$.

The partition induced by $\sim$ is clearly finite, since the order of m_i integers is finite, and the number of multi-level maps D on a finite set is also finite. Let $s := |\mathcal{Z}|$ be the cardinality of the set $\mathcal{Z}$. We show in Cummins et al. (2016) that each $u \in \mathcal{Z}$ has a computable geometrical representation as a connected subset $U \subset P$. Therefore there is a computable decomposition of the parameter space P in s regions U_i for $i = 1, \ldots, s$, such that for any $p, q \in U_i$ we have $D^p = D^q$, and hence also $\mathcal{F}^p = \mathcal{F}^q$. Therefore a finite collection $\{\mathcal{F}^u\}_{u\in\mathcal{Z}}$ captures dynamics of all maps $\mathcal{F}^p$ across all the parameter space P.

We remark that the parameter graph captures the dynamics of all subgraphs of **RN** as well as **RN** itself. Although not addressed in this paper, we can limit the exploration of the dynamics only to those combinatorial parameters that result in **RN**-compatible multi-level discrete maps D.

4. DSGRN: DYNAMICAL SIGNATURES GENERATED BY REGULATORY NETWORKS

Given a network **RN** and the associated switching system, the computational tool DSGRN (Cummins et al., 2016; Harker, 2018) computes and records a graph of graphs in SQL database format. This general database can be queried in many ways, and we will give a short example after defining the graphs that are computed. If a user starts with a synchronous Boolean model B, the first step is to calculate an the interaction graph **RN** of B. DSGRN then describes the long term dynamics of all multi-valued nearest neighbor maps compatible with the switching systems associated to **RN**. Each of these multi-valued nearest neighbor maps is an asynchronous update of a multi-level discrete map. Therefore DSGRN embeds the dynamics of B into a family of multi-level discrete models that are all compatible with the dynamics of a switching system associated to **RN**.

Definition 4.1. The parameter graph $\mathcal{P} = (C, A)$ has nodes C that represent all combinatorial parameters via a bijection $h : C \to \mathcal{Z}$. The non-directed edges $(c, c') \in A$ occur when the difference between $h(c) = (O, D)$ and $h(c') = (O', D')$ is exactly one of the following:

1. there is a swap in the order of one pair of adjacent integers j_k, j_l between O and O', and all other elements remain the same;
2. for exactly one $v \in \mathcal{V}$, $||D(v) - D(v')|| = 1$, and $||D(w) - D'(w)|| = 0$ for all $w \in \mathcal{V} \setminus \{v\}$.

For each $u \in \mathcal{Z}$, there is a representative nearest-neighbor multi-valued discrete map $\mathcal{F}^u$. This map can be viewed as a graph.

Definition 4.2. The *state transition graph (STG)* of a switching system with combinatorial parameter u is the directed graph $(\mathcal{V}, \mathcal{E})$, where the nodes $\mathcal{V}$ were defined previously, and $(v, w) \in \mathcal{E}$ if and only if $w \in \mathcal{F}^u(v)$.

A *recurrent component* (also referred to as a *strongly connected path component*) of the STG $(\mathcal{V}, \mathcal{E})$ is a maximal collection $\mathcal{M}$ of vertices such that for any $u, v \in \mathcal{M}$ there exists a non-empty path from u to v within the subgraph induced by $\mathcal{M}$. The collection of all recurrent components of $(\mathcal{V}, \mathcal{E})$ is denoted by

$$\mathsf{MD}(\mathcal{F}) := \left\{\mathcal{M}(i) \subset \mathcal{V} \mid i \in \mathsf{P}\right\}$$

and is called a *Morse decomposition* of the STG. Here P is an index set. Recurrent components inherit a well-defined partial order by the reachability relation in the directed graph $(\mathcal{V}, \mathcal{E})$. In particular, there is a partial order on the indexing set P of $\mathsf{MD}(\mathcal{F})$ defined by $\mathrm{i} \leq \mathrm{j}$ if there exists a path in $(\mathcal{V}, \mathcal{E})$ from an element of $\mathcal{M}(j)$ to an element of $\mathcal{M}(i)$.

Definition 4.3. The *Morse graph* of the STG, denoted $\mathsf{MG}(\mathcal{F})$, is the Hasse diagram of the poset $(\mathsf{P}, \leq)$. We refer to the elements of P as the *Morse nodes* of the graph.

Any recurrent behavior of the ODE system will be be captured by one of the Morse nodes of the Morse graph. That is, any recurrent set of the ODE will be a subset of a set of domains that correspond to states in STG that belong to a single Morse node.

Each component of the Morse graph can be annotated. We use the following terminology:

1. FP denotes a Morse graph component consisting of a single node of the state transition graph (STG).
2. $\mathsf{FP}(v)$ denotes an FP that is located in $\kappa = g^{-1}(v)$ for $v \in \mathcal{V}$.
3. $\mathsf{FP\ ON}$ denotes an FP in which the associated v has no zeros.
4. $\mathsf{FP\ OFF}$ denotes an FP in which the associated v is all zeros.
5. FC denotes a Morse graph component $\mathcal{M}$ that contains at least one path through the subgraph induced by $\mathcal{M}$ that crosses at least one threshold in each variable x_i. FC stands for "full cycle."
6. $\mathsf{XC}(x_{j_1}, \ldots, x_{j_n})$ denotes a partial oscillation in variables $x_{j_1}, \ldots, x_{j_n}$, where only thresholds in these variables are crossed by paths in the Morse graph component.

If a component is a leaf of the Morse graph, i.e., it has no outgoing edges, then we call it an *attractor*. For each node in the parameter graph, DSGRN records the annotated Morse graph, and this collection comprises the database.

5. EXAMPLE

A DSGRN Database can queried via any general expression in SQL. Some queries have been implemented on a sample set of databases at http://chomp.rutgers.edu/Projects/DSGRN/DB/index.html. See **Figure 1A** for a screenshot of the above website showing networks with precomputed databases. This screenshot shows a selection of different regulatory networks, each of which may be clicked on to show detailed information about the computation of the network dynamics. **Figure 1B** shows a screenshot the result of such a click, and **Figure 1B** shows the result of applying a filter to the network dynamics. We will now step through each of these screenshots in more detail to explain the displayed summary of network dynamics.

In **Figure 1A**, in the third row on the right, we see a network labeled `5D_2015_10_21_VA`. Clicking on it, we see the middle screenshot in **Figure 1B**. The picture of the network **RN** is in the upper left, and next to it an Annotation Filter, which allows us to filter the results based on the annotations of the displayed Morse graphs. All of the annotated Morse graphs that are generated by at least one combinatorial parameter are shown, ordered by the number of combinatorial parameters that produced the given Morse graph. By clicking on the "Yes" button beside FC, we select the Morse graphs that contain a component annotated by FC. In **Figure 1C**, we show a few top Morse graphs satisfying this condition. By choosing different combinations of "Yes", "No", and "Either" in the Annotation Filter, we can explore the different dynamical behaviors of the system.

Although graphical display of the database is useful for exploratory purposes, it is not as powerful as SQL searches over the DSGRN database in which arbitrary combinations of annotated Morse graphs can be selected. Moreover, to use graphical display it is necessary to set up a server. The expected use of DSGRN is to calculate the database and then to use flexible, user-defined SQL queries to search for dynamics of interest.

We now show how to perform some queries that are not available in our demo website. In order to compute the database for DSGRN, the user needs to install DSGRN (Harker, 2018) from GitHub, following the instructions on http://dsgrn.readthedocs.io/en/latest/index.html. While we intend to provide SBML compatibility in the near future, currently the user needs to create a network file that provides names for each node in the regulatory network **RN** and describes the input logic function M_i for each node i. The following is the network file for `5D_2015_10_21_VA` as shown in the upper left of the middle screenshot in **Figure 1B**:

```
p53 : (Chk2 + ATM)(~Mdm2)
ATM : ~Wip1
Chk2 : ATM (~Wip1)
Wip1 : p53
Mdm2 : p53
```

The name of the node is on the left hand side of the colon, and the input logic function M_i to the node is on the right hand side. For example, p53 has three inputs, with "OR" (addition) logic between Chk2 and ATM, and "AND NOT" (multiplication) logic on Mdm2. The symbol "~" denotes repression. Suppose that this file is saved under "RN.txt." To compute the DSGRN SQL Database named "RN.db" using 4 threads we run the following command:

```
mpiexec -np 4 Signatures RN.txt RN.db
```

After the database is computed, we can query RN.db for different dynamical behaviors. Several tables for the database are automatically generated, including Signatures, MorseGraphAnnotations, and MorseGraphEdges, which we will use in queries below. For a comprehensive list of the tables generated, more detail on the SQL database, and other queries, see the links from the documentation site http://dsgrn.readthedocs.io/en/latest/index.html.

We take the number of combinatorial parameters that generates a specific dynamical behavior to be a proxy for the robustness of the behavior across all of parameter space. The number of combinatorial parameters for network **RN** specified in RN.txt is the number of rows in the database RN.db. Therefore we can find the number of parameters using the command:

```
sqlite3 RN.db 'select count(*) from
Signatures'
```

which in this case tells us that there are 803,520 parameters associated to the network `5D_2015_10_21_VA`. We now search the database for the number of combinatorial parameters with at least one *stable* FC. Note that the Annotation Filter in **Figure 1B** searches for any FC, including unstable ones. The command for this search is

```
sqlite3 RN.db 'select count(*) from
Signatures natural join
(select distinct(MorseGraphIndex) from
(select MorseGraphIndex,Vertex from
MorseGraphAnnotations where Label="FC"
except select MorseGraphIndex,Source from
MorseGraphEdges))'
```

and the result is 6904 combinatorial parameters, which is 0.86% of all the parameters. In contrast, the number with at least one stable FP is 667,536, which is 83% of the parameters, obtained by:

```
sqlite3 RN.db 'select count(*) from
Signatures natural join
(select distinct(MorseGraphIndex) from
(select MorseGraphIndex,Vertex from
MorseGraphAnnotations where Label like
"FP%"
except select MorseGraphIndex,Source from
MorseGraphEdges))'
```

Based on the results of these queries, we conclude that a stable FP is far more common that a stable FC, and therefore a more robust behavior for this network.

Table 1 shows the computational scaling of DSGRN in a series of small networks taken from http://chomp.rutgers.edu/Projects/DSGRN/DB/index.html, some of which are shown **Figure 1A**. We see that the computation time and database storage increase

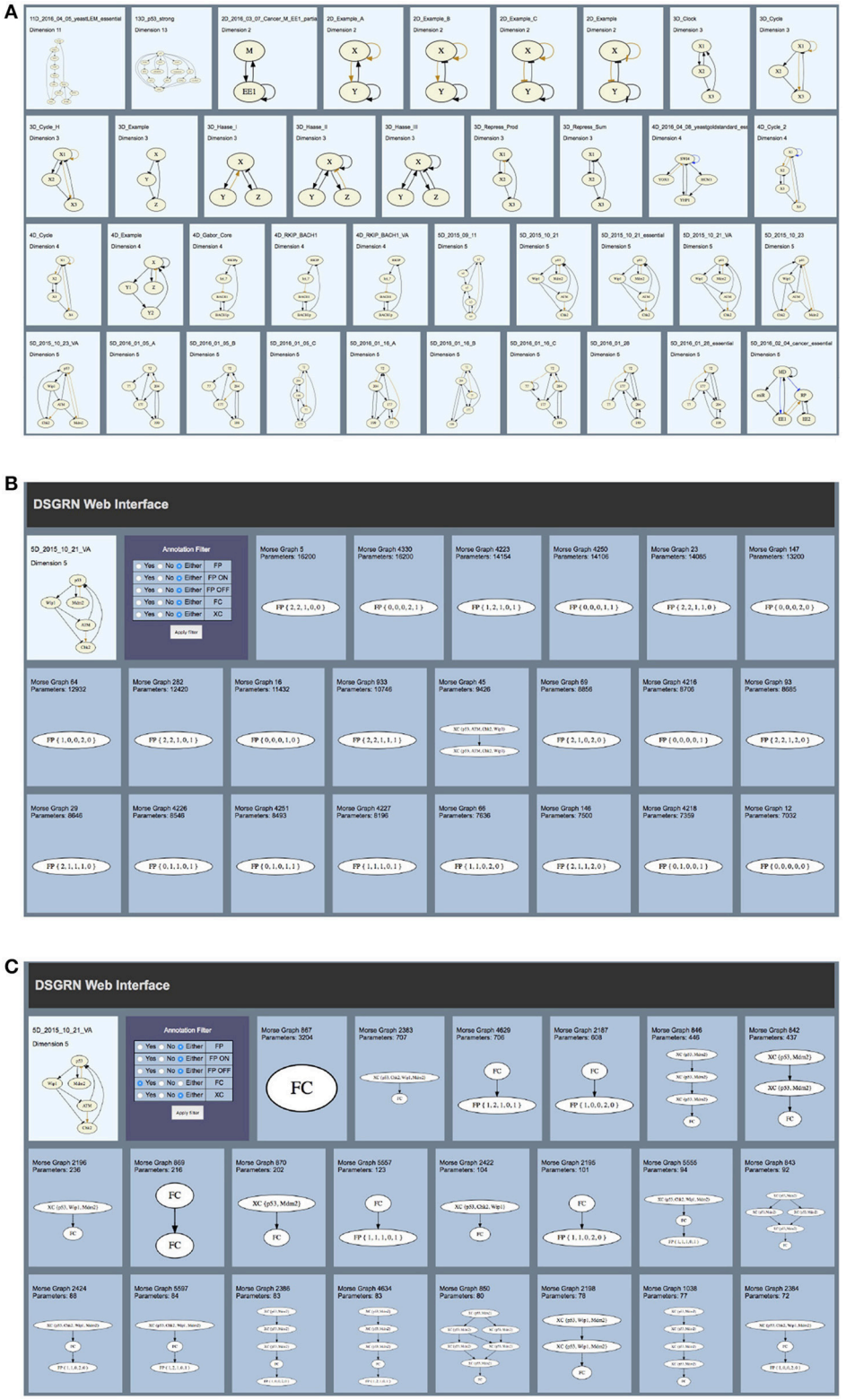


FIGURE 1 | Screenshots of http://chomp.rutgers.edu/Projects/DSGRN/DB/index.html. The description of the Figure and step-by-step guide through an example is in the text.

TABLE 1 | Example performance of DSGRN on 4 threads on a 2013 MacBook Pro. In practice, DSGRN is limited more by storage space than by computation time.

Name	# Nodes	# Edges	# Parameters	Time	Storage
2D_Example_A	2	4	1,600	2.7 s	124 K
3D_Cycle	3	5	5,400	3.1 s	224 K
4D_Example	4	6	122,472	10.4 s	4 M
5D_2015_10_21_VA	5	8	803,520	2 m 26 s	46 M
7D_2016_04_05_yeastLEM	7	10	3,499,200	12 m 41 s	128 M

rapidly as the network size increases. This increase is due particularly to the presence of high degree nodes, rather than to the absolute number of nodes and edges. High degree nodes cause the most rapid increase in the number of combinatorial parameters. Because of parallelization and usage of computing clusters with a large core count, we find in practice that DSGRN is more limited by space to store databases than by computation time.

In order to address the storage space scaling limitations, we have implemented two additions to DSGRN. The first is the idea of "essential" parameters, which is the subset of parameters consistent with Definition 2.2. DSGRN was originally designed to study not only **RN**-compatible asynchronous multi-level maps, but all such maps that were **S**-compatible with any subgraph **S** of **RN**. By limiting ourselves to **RN**-compatible maps, the size of parameter space is greatly reduced. To specify essential parameters, add "`: E`" to the end of every line in the network specification file for **RN**. For example, the essential network specification file for `2D_Example_A` using multiplicative logic is:

```
X : XY : E
Y : XY : E
```

The second addition is an extensive Python module DSGRN that can be used to explore individual parameters rather than calculating the entire database at once. This model is part of the standard DSGRN installation. If a hypothesis about the network dynamics can be constructed a priori, then the selection for annotated Morse graphs can be computed on the fly, allowing much larger networks to be analyzed than is otherwise possible. See https://github.com/shaunharker/DSGRN/blob/master/Tutorials/GettingStarted.ipynb for a brief introduction to the Python library.

6. DISCUSSION

Given a regulatory network **RN** there is a very large number of multi-level maps D that can be associated to this network. We can enumerate them by selecting for each node an arbitrary assignment of node value based on the node inputs. If the structure of the network is the only information available, these all represent valid models for the network dynamics in the class of discrete multi-level maps, which generalize Boolean models. This class of functions generate, via asynchronous update, a class of multi-valued nearest neighbor maps $\mathcal{F}$ which better represent biological reality. States of $\mathcal{F}$ only change one at a time.

To make the collection of **RN**-compatible functions $\mathcal{F}$ smaller and more biologically realistic, we employ a switching system, which is an ODE system with discrete-valued interaction terms. They were introduced in the 1970's (Glass and Kauffman, 1972, 1973) as a continuous time counterpart to Boolean networks. A switching system is parameterized by continuous parameters, but this set decomposes into a finite number of computable regions (Cummins et al., 2016), each of which is associated with a single multi-level map D^u and its asynchronous update $\mathcal{F}^u$, where $\mathcal{F}^u$ is compatible with the switching system ODE (Crawford-Kahrl et al., 2018). The mutual position of these regions in the parameter space provide a natural way to define a notion of "neighboring" functions D^u, D^v (and thus $\mathcal{F}^u, \mathcal{F}^v$).

Our computational tool DSGRN (Cummins et al., 2016; Cummins et al., 2017; Harker, 2018) constructs the collection of all such parameter regions and encodes them in the form of a parameter graph. For each node u of the parameter graph, the DSGRN Database stores information about the global dynamics in form of a Morse graph, which is a summary of the dynamics of $\mathcal{F}^u$. A DSGRN Database provides a summary of dynamics for all maps $\mathcal{F}^u$ which are compatible with a switching system on **RN**. In this sense DSGRN represents the dynamics compatible with the network **RN** across all parameters.

DSGRN can be used to either list dynamical behaviors that are compatible with a given network **RN**, or search in the space of networks for those networks that provide most robustly dynamics of interest, for instance FC or FP.

AUTHOR CONTRIBUTIONS

TG, KM conceptualized the paper. TG, BC wrote the paper. SH, BC implemented the methods and performed computations.

ACKNOWLEDGMENTS

TG was partially supported by NSF grants DMS-1226213, DMS-1361240, USDA 2015-51106-23970, DARPA grants D12AP200025 and FA8750-17-C-0054, and NIH grants 1R01AG040020-01 and 1R01GM126555-01. BC was partially supported by grants USDA 2015-51106-23970, DARPA grants D12AP200025 and FA8750-17-C-0054 and NIH 1R01GM126555-01. The work of SH and KM was partially supported by grants NSF-DMS-1125174, 1248071, 1521771, NIH 1R01GM126555-01 and DARPA contracts HR0011-16-2-0033, FA8750-17-C-0054. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

REFERENCES

Albert, R. (2007). Network inference, analysis, and modeling in systems biology. *Plant Cell* 19, 3327–3338. doi: 10.1105/tpc.107.054700

Albert, R., Collins, J. J., and Glass, L. (2013). Introduction to focus issue: quantitative approaches to genetic networks. *Chaos* 23:025001. doi: 10.1063/1.4810923

Batt, G., Belta, C., and Weiss, R. (2007a). "Model checking genetic regulatory networks with parameter uncertainty," in *Hybrid Systems: Computation and Control, HSCCÔ07, Lecture Notes in Computer Science 4416* (Berlin: Springer), 61–75.

Batt, G., Yordanov, B., Weiss, R., and Belta, C. (2007b). Robustness analysis and tuning of synthetic gene networks. *Bioinformatics* 23, 2415–2422. doi: 10.1093/bioinformatics/btm362

Belta, C., and Habets, L. (2006). Controlling a class of nonlinear systems on rectangles. *Trabs. Aut. Control* 51, 1749–1759. doi: 10.1109/TAC.2006.884957

Bernard, O., and Gouze, J. (2002). Global qualitative description of a class of nonlinear dynamical systems. *Artif. Intell.* 136, 29–59. doi: 10.1016/S0004-3702(01)00169-2

Bornholt, S. (2008). Boolean network models of cellular regulation: prospects and limitations. *J. R. Soc. Interface* 5, 134–150. doi: 10.1098/rsif.2008.0132.focus

Chaves, M., Sontag, E. D., and Albert, R. (2006). Methods of robustness analysis for Boolean models of gene control networks. *IEE Proc. Syst. Biol.* 153, 154–167. doi: 10.1049/ip-syb:20050079

Chen, K., Calzone, L., Csikasz-Nagy, A., Cross, F., Novak, B., and Tyson, J. (2004). Integrative analysis of cell cycle control in budding yeast. *Mol. Biol. Cell* 15, 3841–3862. doi: 10.1091/mbc.e03-11-0794

Crawford-Kahrl, P., Cummins, B., and Gedeon, T. (2018). Comparison of two combinatorial models of global network dynamics. arXiv 1801.06524.

Cummins, B., Gedeon, T., Harker, S., and Mischaikow, K. (2017). "Database of dynamic signatures generated by regulatory networks (DSGRN)," in *Computational Methods in Systems Biology - 2017*, eds J. Feret and H. Koeppl (Cham: Springer), 300–308. Available online at: https://www.springer.com/us/book/9783319674704

Cummins, B., Gedeon, T., Harker, S., Mischaikow, K., and Mok, K. (2016). Combinatorial representation of parameter space for switching systems. *SIAM J. Appl. Dyn. Syst.* 15, 2176–2212. doi: 10.1137/15M1052743

de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: a literature review. *J. Comput. Biol.* 9, 67–103. doi: 10.1089/10665270252833208

de Jong, H., Gouze, J., Hernandez, C., Page, M., Sari, T., and Geiselmann, J. (2004). Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bull. Math. Biol.* 66, 301–340. doi: 10.1016/j.bulm.2003.08.010

Edwards, R. (2001). Chaos in neural and gene networks with hard switching. *Diff. Eq. Dyn. Sys.* 9, 187–220.

Edwards, R., Machina, a., McGregor, G., and van den Driessche, P. (2015). A modelling framework for gene regulatory networks including transcription and translation. *Bull. Math. Biol.* 77, 953–983. doi: 10.1007/s11538-015-0073-9

Faure, A., Naldi, A., Chaouiya, C., , and Thieffry, D. (2006). Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle. *Bioinformatics* 22, e124–e131. doi: 10.1093/bioinformatics/btl210

Glass, L., and Kauffman, S. a. (1972). Co-operative components, spatial localization and oscillatory cellular dynamics. *J. Theor. Biol.* 34, 219–37. doi: 10.1016/0022-5193(72)90157-9

Glass, L., and Kauffman, S. a. (1973). The logical analysis of continuous, non-linear biochemical control networks. *J. Theor. Biol.* 39, 103–129. doi: 10.1016/0022-5193(73)90208-7

Goncalves, E., Bucher, J., Ryll, A., Niklas, J., Mauch, K., Klamt, S., et al. (2013). Bridging the layers: towards integration of signal transduction, regulation and metabolism into mathematical models. *Mol. Biosyst.* 9, 1576–1583. doi: 10.1039/c3mb25489e

Harker, S. (2018). *DSGRN Software*. Available online at: https://github.com/shaunharker/DSGRN

Heatha, A., and Kavria, L. (2009). Computational challenges in systems biology. *Comput. Sci. Rev.* 3, 1–17. doi: 10.1016/j.cosrev.2009.01.002

Ironi, L., Panzeri, L., Plahte, E., and Simoncini, V. (2011). Dynamics of actively regulated gene networks. *Physica D* 240, 779–794. doi: 10.1016/j.physd.2010.12.010

Karlebach, G., and Shamir, R. (2008). Modelling and analysis of gene regulatory networks. *Nature* 9:770. doi: 10.1038/nrm2503

Machado, D., Costa, R., Rocha, M., Ferreira, E., Tidor, B., and Rocha, I. (2011). Modeling formalisms in systems biology. *AMB Exp.* 1:45. doi: 10.1186/2191-0855-1-45

Pauleve, L., and Richard, A. (2012). Static analysis of boolean networks based on interaction graphs: a survey. *Electr. Notes Theor. Comput. Sci.* 284, 93–104. doi: 10.1016/j.entcs.2012.05.017

Saadatpour, A., and Reka, A. (2013). Boolean modeling of biological regulatory networks: a methodology tutorial. *Methods* 62, 3–12. doi: 10.1016/j.ymeth.2012.10.012

Thomas, R. (1973). Boolean formalization of genetic control circuits. *J. Theor. Biol.* 42, 563–585. doi: 10.1016/0022-5193(73)90247-6

Thomas, R., Thieffry, D., and Kaufman, M. (1995). Dynamical behaviour of biological regulatory networks-i. biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bull. Math. Biol.* 57, 247–276. doi: 10.1007/BF02460618

Tournier, L., and Chaves, M. (2009). Uncovering operational interactions in genetic networks using asynchronous boolean dynamics. *J. Theor. Biol.* 260, 196–209. doi: 10.1016/j.jtbi.2009.06.006

Tyson, J. J., and Novak, B. (2013). "Chapter 14 - irreversible transitions, bistability and checkpoint controls in the eukaryotic cell cycle: a systems-level understanding," in *Handbook of Systems Biology*, ed A. M. W. V. Dekker (San Diego, CA: Academic Press), 265–285.

von Dassow, G., Meir, E., Munro, E., and Odell, G. (2000). The segment polarity network is a robust development module. *Nature* 406, 188–192. doi: 10.1038/35018085

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

ORIGINAL RESEARCH
published: 29 May 2018
doi: 10.3389/fphys.2018.00586

Analysis Tools for Interconnected Boolean Networks With Biological Applications

Madalena Chaves[1]* *and Laurent Tournier*[2]

[1] Inria Sophia Antipolis - Méditerranée, Université Côte d'Azur, Valbonne, France, [2] MaIAGE, INRA, Université Paris-Saclay, Jouy-en-Josas, France

Boolean networks with asynchronous updates are a class of logical models particularly well adapted to describe the dynamics of biological networks with uncertain measures. The state space of these models can be described by an asynchronous state transition graph, which represents all the possible exits from every single state, and gives a global image of all the possible trajectories of the system. In addition, the asynchronous state transition graph can be associated with an absorbing Markov chain, further providing a semi-quantitative framework where it becomes possible to compute probabilities for the different trajectories. For large networks, however, such direct analyses become computationally untractable, given the exponential dimension of the graph. Exploiting the general modularity of biological systems, we have introduced the novel concept of *asymptotic graph*, computed as an interconnection of several asynchronous transition graphs and recovering all asymptotic behaviors of a large interconnected system from the behavior of its smaller modules. From a modeling point of view, the interconnection of networks is very useful to address for instance the interplay between known biological modules and to test different hypotheses on the nature of their mutual regulatory links. This paper develops two new features of this general methodology: a quantitative dimension is added to the asymptotic graph, through the computation of relative probabilities for each final attractor and a companion *cross-graph* is introduced to complement the method on a theoretical point of view.

Keywords: asynchronous Boolean networks, module interconnection, state transition graph, attractor computation, biological regulatory networks

OPEN ACCESS

Edited by:
Matteo Barberis,
University of Amsterdam, Netherlands

Reviewed by:
Laurence Calzone,
Institut Curie, France
Olaf Wolkenhauer,
University of Rostock, Germany
Aurélien Naldi,
École Normale Supérieure, France

***Correspondence:**
Madalena Chaves
madalena.chaves@inria.fr

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: *01 February 2018*
Accepted: *02 May 2018*
Published: *29 May 2018*

Citation:
Chaves M and Tournier L (2018) Analysis Tools for Interconnected Boolean Networks With Biological Applications. Front. Physiol. 9:586. doi: 10.3389/fphys.2018.00586

1. INTRODUCTION

An intuitive representation of system interactions, an algorithmic description of state transitions, and the capacity to capture the global dynamics of the system, list some of the advantages of Boolean models, which remain a powerful tool in the modeling and analysis of biological networks (Wang et al., 2012; Abou-Jaoudé et al., 2016). Successfully predictive examples of Boolean models cover complex networks across many different organisms, from cell cycle (Li et al., 2004; Fauré et al., 2006), to fly or plant morphogenesis (Albert and Othmer, 2003; García-Gómez et al., 2017), and highly complex networks such as T-cell induction (Mendoza and Xenarios, 2006; Saez-Rodriguez et al., 2007), leukemia (Zhang et al., 2008) or apoptosis (Calzone et al., 2010).

In a modular view of a biological organism, each task is executed by a specific set of interactions among an ensemble of biological components; in other words, it can be said that there is a specifc network, or module, for each specific task (signaling, metabolic, physiological, etc.). These modules often interact with each other, one task triggering the next in a chain of events or cyclic phenomena. Examples include chains of signaling networks such as MAPK cascades, genetic-metabolic interactions (Baldazzi et al., 2010), or coupled oscillations (Gérard and Goldbeter, 2012). However, in many cases, while experimental evidence supports the existence of links between two modules, their modes of interaction are still unclear (as in the case of mammalian cell cycle and circadian clock, see Feillet et al., 2015). In this context, mathematical tools are necessary to facilitate the analysis of the complex behavior obtained from the interconnection of two or more known modules.

One of the challenges in the analysis of Boolean networks is attractor computation, particularly for high-dimensional networks. For a network of dimension n, the size of the state transition graph is 2^n. A direct analysis of such a graph may become computationally costly, in terms of space and time, when $n \geq 20$. This is especially true with asynchronous updating, which includes numerous dynamical trajectories. Two very efficient methods have recently been developed: Zañudo and Albert (2013) compute all attractors of a network (up to $n \approx 100$), by isolating special properties of the state transition graph's components; Veliz-Cuba et al. (2014) compute all singletons (attractors containing a single state) for networks up to $n = 1{,}000$, by using a computational algebra approach.

In this paper, we propose a methodology aimed specifically at analyzing the interconnection between several known Boolean modules. The interconnection between two biological networks can be very hard to test *in vivo*: our methodology provides a platform for hypothesis testing, confirming or disproving assumptions regarding mutual regulatory effects, simulating and comparing various forms of interconnection schemes and corresponding emergent dynamical behavior. Our method relies on the construction of a new object, *the asymptotic graph*, introduced by Tournier and Chaves (2013), which is a directed graph constructed only from the set of attractors of each module and that captures all the asymptotic behaviors of the interconnected network.

After a brief review of Boolean network interconnections, two improvements to the asymptotic graph are introduced in this paper, to mitigate two of its known limitations. First, it was observed that the asymptotic graph may also recover spurious attractors, in addition to the true attractors of the full network (Tournier and Chaves, 2013); we introduce an extension, called the *cross graph* that solves this issue from a theoretical point of view. The cross graph is constructed from the set of strongly connected components of each separate module, while the asymptotic graph is constructed from *terminal* strongly connected components only. Second, to enrich the traditional ON/OFF representation inherent to Boolean models, we propose a method to assign probabilities to the edges of the asymptotic graph, thereby allowing a probabilistic representation of the various possible trajectories of the composed network. Our methodology is applied first to a class of general randomly generated Boolean models and then to two state-of-the-art biological models in two different organisms: (i) to explore the interplay between mammalian cell cycle and circadian clock oscillators and (ii) to test hypotheses on the regulatory links between budding yeast cell cycle and cell size, where our analysis suggests that the START signal should come from mitosis phase.

2. INTERCONNECTIONS OF ASYNCHRONOUS BOOLEAN NETWORKS: A SHORT REVIEW

Throughout this paper, we will consider Boolean networks under asynchronous updates. An interconnected Boolean network is, briefly, the combined network formed by linking together, in an approriately prescribed way, two or more separate Boolean modules. In previous works (Chaves and Tournier, 2011; Tournier and Chaves, 2013) we have introduced a new object, the *asymptotic graph*, that characterizes the attractors of the combined Boolean network in terms only of the attractors of the separate modules—hence with no need to compute the larger state transition graph. In the following, the definition of the main objects needed to introduce the asymptotic graph are briefly reviewed.

2.1. IO Asynchronous Boolean Networks and Their Interconnections

Let us start by a brief recall of the definition of an input-output asynchronous Boolean network (IO ABN), reprising the notation introduced by Tournier and Chaves (2013). An IO ABN Σ^A is characterized by three integers n_A, p_A, q_A ($n_A > 0$ is the dimension of the system, $p_A, q_A \geq 0$ are respectively the numbers of inputs and outputs) and by two Boolean maps: $f^A : \{0,1\}^{p_A} \times \{0,1\}^{n_A} \to \{0,1\}^{n_A}$ (the transition function) and $h^A : \{0,1\}^{n_A} \to \{0,1\}^{q_A}$ (the output function). For any given input profile $u \in \{0,1\}^{p_A}$, the asynchronous dynamics of the network are given by the *asynchronous transition graph* $G^{A,u}$, which is a digraph over the vertex set $\{0,1\}^{n_A}$ defined as follows: for any state $x = (x_1, \ldots, x_n) \in \{0,1\}^{n_A}$, the set of its successors are the states $(x_1, \ldots, \neg x_i, \ldots, x_n)$, for all $i \in \{1, \ldots, n\}$ such that $f_i^A(u, x) \neq x_i$. The number of vertices of such a graph is 2^{n_A} and its number of arcs, denoted by m_A, verifies $0 \leq m_A \leq n_A 2^{n_A}$. It is therefore relatively sparse and can thus be efficiently stored by a $2^{n_A} \times 2^{n_A}$ adjacency matrix. In the following, we will consider that $G^{A,u}$ designates this matrix. Given two integers $i, j \in \{1, \ldots, 2^{n_A}\}$, the (i, j) entry of the adjacency matrix equals 1 if state j is a successor of state i and 0 otherwise. In a classical abuse of notation, we associate each integer $i \in \{1, \ldots, 2^{n_A}\}$ with its binary representation $x \in \{0,1\}^{n_A}$ in lexicographic order, with the left-most bit being the most significant one; in other words: $i - 1 = \sum_{k=1}^{n_A} x_k 2^{n_A - k}$. Thus, we will indifferently call *state* either an integer $i \in \{1, \ldots, 2^{n_A}\}$ or its Boolean representation $x \in \{0,1\}^{n_A}$.

EXAMPLE 1. Consider the bidimensional single-input, single-output (SISO) network defined by: $f^A(u, x_1, x_2) = (u, x_1)$ and

$h^A(x_1, x_2) = x_2$. Graphically, this network can be represented as a simple cascade $u \to x_1 \to x_2$. Its dynamics are characterized by the two graphs $G^{A,0}$ and $G^{A,1}$, represented below in graphical and matricial forms:

$$G^{A,0}: \begin{array}{ccc} 10 & \to & 11 \\ \downarrow & & \downarrow \\ 00 & \leftarrow & 01 \end{array} \quad \begin{pmatrix} 0&0&0&0 \\ 1&0&0&0 \\ 1&0&0&1 \\ 0&1&0&0 \end{pmatrix},$$

$$G^{A,1}: \begin{array}{ccc} 01 & \to & 00 \\ \downarrow & & \downarrow \\ 11 & \leftarrow & 10 \end{array} \quad \begin{pmatrix} 0&0&1&0 \\ 1&0&0&1 \\ 0&0&0&1 \\ 0&0&0&0 \end{pmatrix}.$$

In adjacency matrices, by convention the (i, j) entry equals 1 iff state j is a successor of state i. Here, the four states (rows and columns of the matrix) are intended in the following order: 00, 01, 10, 11. In $G^{A,0}$, state 00 does not have any successor, implying the first row of its adjacency matrix is zero: 00 is a steady state of the network. Similarly, 11 is a steady state of $G^{A,1}$. □

Classically, an asynchronous transition graph $G^{A,u}$ is analyzed by first computing its decomposition into strongly connected components (SCCs), denoted by $A^1_u, \ldots, A^{N^A_u}_u$, where $1 \leq N^A_u \leq 2^n$. The set of all SCCs forms a partition of the state space $\{0,1\}^{n_A}$ and their computation can be efficiently achieved in $O(2^{n_A} + m_A)$. By contracting each SCC to a single vertex, a directed acyclic graph (dag) is constructed, sometimes called *condensation* graph or simply SCC graph. This dag provides a useful description of key dynamical behaviors of the network; in particular terminal SCCs (the leafs of the dag) correspond to the attractors of the network. More details about these graph theoretical tools can be found, for instance, in the textbook by Cormen et al. (2001).

Consider now two IO ABN Σ^A and Σ^B, of respective dimensions (n_A, p_A, q_A) and (n_B, p_B, q_B) and state variables $x \in \{0,1\}^{n_A}$ and $y \in \{0,1\}^{n_B}$. Note that all the methods presented in this paper generalize to more than two modules; however, in order to maintain a clear exposition of the results, the definitions are given for interconnections of two modules. An *interconnection scheme* of Σ^A and Σ^B consists in two interconnecting functions $\mu_A : \{0,1\}^{q_B} \to \{0,1\}^{p_A}$ and $\mu_B : \{0,1\}^{q_A} \to \{0,1\}^{p_B}$ mapping the outputs of each module to the inputs of the other module. For convenience, throughout this paper we will make the assumption that $q_B = p_A$ and $q_A = p_B$ and that the interconnecting functions are simply identity maps. Following Tournier and Chaves (2013), with this assumption the resulting interconnected network is the ABN of dimension $n_A + n_B$, with no input and no output, defined by the following transition function:

$$\begin{array}{rccc} f: & \{0,1\}^{n_A} \times \{0,1\}^{n_B} & \longrightarrow & \{0,1\}^{n_A} \times \{0,1\}^{n_B} \\ & (x, y) & \longmapsto & \left(f^A(h^B(y), x), f^B(h^A(x), y)\right). \end{array} \quad (1)$$

One can then consider the interconnection as a standalone network: its transition graph G can be constructed from this transition function f. Alternatively, one can also build the graph G directly from the set of transition graphs $G^{A,u}$, $u \in \{0,1\}^{p_A}$ and $G^{B,\upsilon}$, $\upsilon \in \{0,1\}^{p_B}$ as follows. Let (x, y) and (x', y') be two Boolean vectors in $\{0,1\}^{n_A} \times \{0,1\}^{n_B}$, then (x', y') is a (asynchronous) successor of (x, y) if

- either $x = x'$ and y' is a successor of y in $G^{B,h^A(x)}$,
- or $y = y'$ and x' is a successor of x in $G^{A,h^B(y)}$.

It is possible to summarize this definition in a simple matricial form. First, for each $\alpha \in \{0,1\}^{q_A}$, introduce the $2^{n_A} \times 2^{n_A}$ diagonal Boolean matrix $\Delta^{A,\alpha}$ such that $\left[\Delta^{A,\alpha}\right]_{ii} = 1$ if the output of state i is equal to α and 0 otherwise. Similarly, for module Σ^B introduce the $2^{n_B} \times 2^{n_B}$ diagonal Boolean matrices $\Delta^{B,\beta}$, with $\beta \in \{0,1\}^{q_B}$. Then, G can be reconstructed by the formula:

$$G := \bigvee_{(\alpha,\beta) \in \{0,1\}^{q_A} \times \{0,1\}^{q_B}} \left(G^{A,\beta} \otimes \Delta^{B,\beta} \vee \Delta^{A,\alpha} \otimes G^{B,\alpha}\right), \quad (2)$$

where $\otimes$ designates the classical Kronecker product. By replacing matrices Δ with identity matrices, one may recognize in this definition of G the notion of Cartesian product of graphs, first introduced by Sabidussi (1959). To be more precise, (2) generalizes the notion of Cartesian product to interconnections, by including only transitions that are consistent with the input-output scheme.

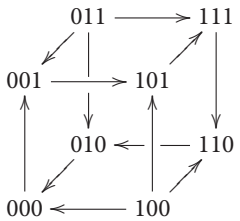
EXAMPLE 2. Consider module Σ^A defined in Example 1 and let the one-dimensional SISO module Σ^B defined by $f^B(\upsilon, y_1) = \neg\upsilon$ and $h^B(y_1) = y_1$. Its dynamics are given by

$$G^{B,0} = \begin{pmatrix} 0&1 \\ 0&0 \end{pmatrix}, \text{ and } G^{B,1} = \begin{pmatrix} 0&0 \\ 1&0 \end{pmatrix}.$$

The interconnected network can be reconstructed by using (1), leading to the 3-dimensional transition function $f(x_1, x_2, y_1) = (y_1, x_1, \neg x_2)$. Alternatively, the transition graph G can also be computed directly as the interconnection of the dynamics of the two separated modules by using (2):

$$\begin{aligned} G &= \left(G^{A,0} \otimes \Delta^{B,0}\right) \vee \left(G^{A,1} \otimes \Delta^{B,1}\right) \vee \left(\Delta^{A,0} \otimes G^{B,0}\right) \vee \left(\Delta^{A,1} \otimes G^{B,1}\right), \\ &= \begin{pmatrix} 0&0&0&0 \\ 1&0&0&0 \\ 1&0&0&1 \\ 0&1&0&0 \end{pmatrix} \otimes \begin{pmatrix} 0&0 \\ 0&1 \end{pmatrix} \vee \begin{pmatrix} 0&0&1&0 \\ 1&0&0&1 \\ 0&0&0&1 \\ 0&0&0&0 \end{pmatrix} \otimes \begin{pmatrix} 1&0 \\ 0&0 \end{pmatrix} \vee \\ &\quad \begin{pmatrix} 1&0&0&0 \\ 0&0&0&0 \\ 0&0&1&0 \\ 0&0&0&0 \end{pmatrix} \otimes \begin{pmatrix} 0&1 \\ 0&0 \end{pmatrix} \vee \begin{pmatrix} 0&0&0&0 \\ 0&1&0&0 \\ 0&0&0&0 \\ 0&0&0&1 \end{pmatrix} \otimes \begin{pmatrix} 0&0 \\ 1&0 \end{pmatrix}, \\ &= \begin{pmatrix} 0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 1&0&0&0&0&0&0&0 \\ 0&1&1&0&0&0&0&1 \\ 1&0&0&0&0&1&1&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \end{pmatrix}. \end{aligned}$$

In graphical form, this transition graph G of the interconnected network can be represented as:



This graph has a unique attractor, composed of six states: $\{001, 101, 111, 110, 010, 000\}$. □

In the present paper, note that we assume the modules and the interconnection scheme are given. It is also possible to consider interconnections as a general *model reduction* technique, where a large network is first decomposed into *a priori* unknown sub-networks. The identification of an efficient decomposition, with the corresponding interconnecting scheme, would then become critical. This problem is related to the general problem of graph partitioning and is addressed elsewhere (Tournier and Chaves, 2013).

2.2. The Asymptotic Graph of an Interconnection

We can now give the definition of the asymptotic graph (Tournier and Chaves, 2013). First, list all the terminal SCCs of module Σ^A: $\left\{A^i_u,\ u \in \{0,1\}^{p_A}, 1 \le i \le L^A_u\right\}$ and cut them with respect to their outputs, *ie.* define, for each output profile $\alpha \in \{0,1\}^{q_A}$, the set $A^i_{u\alpha} := \left\{x \in A^i_u,\ h^A(x) = \alpha\right\}$. For some α such a set may be empty, in that case we will simply omit it. Similarly, define $\left\{B^j_{\upsilon\beta},\ \upsilon \in \{0,1\}^{p_B}, \alpha \in \{0,1\}^{q_B}, 1 \le j \le L^B_\upsilon\right\}$ for module Σ^B. The *asymptotic graph* of the interconnection is then defined as the directed graph $G^{as} = (V^{as}, E^{as})$ such that the vertex set V^{as} is composed of all the cross products $A^i_{u\alpha} \times B^j_{\upsilon\beta}$ and the arc set E^{as} is constructed as follows:

- $A^i_{u\alpha} \times B^j_{\nu\beta} \to A^{i'}_{\beta\alpha'} \times B^j_{\nu\beta}$ iff there exist $x \in A^i_{u\alpha}$, $x' \in A^{i'}_{\beta\alpha'}$ such that there exists a path from x to x' in $G^{A,\beta}$,
- $A^i_{u\alpha} \times B^j_{\nu\beta} \to A^i_{u\alpha} \times B^{j'}_{\alpha\beta'}$ iff there exist $y \in B^j_{\nu\beta}, y' \in B^{j'}_{\alpha\beta'}$ such that there exists a path from y to y' in $G^{B,\alpha}$.

Finally, introduce the function π as follows: if $V = A^i_{u\alpha} \times B^j_{\upsilon\beta} \in V^{as}$, $\pi(V) := \{(x,y), x \in A^i_{u\alpha}, y \in B^j_{\nu\beta}\}$ and if $R \subseteq V^{as}$, $\pi(R) := \bigcup_{V\in R} \pi(V)$. The interest of the asymptotic graph lies in the following theorem, a proof of which can be found in Tournier and Chaves (2013).

THEOREM 1. *If Q is an attractor of the interconnected network, then there exists a terminal SCC R of G^{as} such that $\pi(R) \subseteq Q$.*

EXAMPLE 3. Consider the interconnection of Example 2 above. The asymptotic graph is given by

$$\begin{array}{ccc} A^1_{00}\times B^1_{01} & \to & A^1_{11}\times B^1_{01} \\ \uparrow & & \downarrow \\ A^1_{00}\times B^1_{10} & \to & A^1_{11}\times B^1_{10} \end{array} \quad \text{with:} \quad \begin{cases} \pi(A^1_{00}\times B^1_{01}) = \{001\}, \\ \pi(A^1_{11}\times B^1_{01}) = \{111\}, \\ \pi(A^1_{11}\times B^1_{10}) = \{000\}, \\ \pi(A^1_{00}\times B^1_{10}) = \{110\}. \end{cases}$$

Therefore, G^{as} is composed of a single terminal SCC R, and $\pi(R) = \{001, 111, 000, 110\}$ is actually included into the (unique) attractor of the interconnected network. □

Thanks to Theorem 1, the asymptotic graph is a powerful analytic tool as it recovers *all* the attractors of an interconnection (without missing any), by constructing a graph significantly smaller than the full interconnected graph G (section 4 below provides numerical results for random interconnections). However, it may happen that some terminal SCC of G^{as} does not correspond to an actual attractor of the interconnection. Such terminal SCCs, called *spurious* attractors, appear very rarely and there exist some sufficient conditions to detect *a priori* spurious attractors in certain cases. The most simple one, particularly useful for biological applications is the fact that when R is a singleton then it cannot be a spurious attractor. The proof, along with additional conditions are provided elsewhere (Tournier and Chaves, 2013; Chaves and Carta, 2015).

3. NEW ANALYSIS TOOLS

This section describes our new contributions. Our first goal is to improve the asymptotic graph construction to avoid the generation of spurious attractors (section 3.1) and our second goal is to update the asymptotic graph by adding quantitative information (probabilistic) on the state transitions (section 3.2).

3.1. A Theoretical Tool to Recover All the Dynamics of an Interconnection

The asymptotic graph of an interconnection is constructed only from the modules' attractors, generally implying a relatively manageable size allowing to analyze a wide range of practical examples of interconnections (see sections 4 and 5). Nevertheless, ignoring transient dynamical behaviors of the modules also implies two drawbacks for Theorem 1. First, spurious attractors may appear, although this phenomenon seems to be relatively rare as illustrated in section 4. Second, when a terminal SCC of G^{as} corresponds to an actual attractor, Theorem 1 only ensures an inclusion, meaning the predicted attractor may contain only a small proportion of states that are in the real attractor. We now propose a new graph, called the *cross-graph*, overcoming those two issues and ensuring, at the price of a higher computational cost, a one-to-one recovery of all the attractors of the interconnected network. Note that Tournier and Chaves (2013) already introduced a notion of cross-graph, however the cross-graph described in the following is significantly improved. In particular, its size is bounded by the size of the full interconnected graph, which was not the case for the older version.

Let Σ^A and Σ^B be two IO ABN of respective dimensions (n_A, p_A, q_A) and (n_B, p_B, q_B). As before, suppose for convenience that $p_A = q_B$, $p_B = q_A$ and the interconnecting maps are simply identity maps. We also assume that each module has been separately analyzed: the transition graphs $G^{A,u}$, $u \in \{0,1\}^{p_A}$ and $G^{B,\upsilon}$, $\upsilon \in \{0,1\}^{p_B}$ have been constructed and decomposed into strongly connected components $\left\{A^i_u,\ 1 \le i \le N^A_u\right\}$ for each $u \in \{0,1\}^{p_A}$ and $\left\{B^j_\upsilon,\ 1 \le j \le N^B_u\right\}$ for each $\upsilon \in \{0,1\}^{p_B}$. Let G

denote the full transition graph of the interconnected network, of size $2^{n_A+n_B}$. It can be computed thanks to (2), by interconnecting the modules' transition graphs. The idea behind the cross-graph is to generalize formula (2) in order to interconnect directly the SCCs of those graphs instead of the whole graphs themselves, thus potentially saving a significant amount of space when constructing the dynamics of the interconnection.

First, observe that the strongly connected components $\left\{A_u^i,\ 1 \leq i \leq N_u^A\right\}$ form a partition of the state space $\{0,1\}^{n_A}$ of module Σ^A (N_u^A are integers verifying $1 \leq N_u^A \leq 2^{n_A}$). Therefore, for u varying in $\{0,1\}^{p_A}$ we obtain 2^{p_A} partitions of the same finite set $\Omega = \{0,1\}^{n_A}$. Let $\mathfrak{P}_\Omega$ denote the set of all partitions of Ω. Given two partitions $P_1, P_2 \in \mathfrak{P}_\Omega$, P_1 is said *finer than* P_2, denoted by $P_1 \leq P_2$ if, for each element p in P_1 there is an element q in P_2 such that $p \subseteq q$ (in other words, partition P_1 is a *fragmentation* of partition P_2). The set $(\mathfrak{P}_\Omega, \leq)$ has the structure of a geometric lattice (see *eg.* Birkhoff, 1940). Consequently, for any set $S \subseteq \mathfrak{P}_\Omega$, there exists a (unique) greatest lower bound of S denoted by $\bigwedge S \in \mathfrak{P}_\Omega$. Coming back to the SCC decompositions, introduce the following partition:

$$\begin{aligned} Z^A :&= \bigwedge_{u\in\{0,1\}^{p_A}} \left\{A_u^i,\ 1 \leq i \leq N_u^A\right\}, \\ &= \left\{\mathbf{A}^1, \ldots, \mathbf{A}^{N^A}\right\}, \end{aligned}$$

which is the coarsest partition of $\{0,1\}^{n_A}$ that is finer than every SCC decomposition of all the transition graphs $G^{A,u}$. Once this partition is constructed, following the same idea as before it is further refined by cutting each set $\mathbf{A}^i$ according to their outputs: $\mathbf{A}_\alpha^i := \left\{x \in \mathbf{A}^i,\ h^A(x) = \alpha\right\}$, with the convention that such sets are simply omitted when they are empty. Therefore, we finally obtain a partition $Z_h^A = \left\{\mathbf{A}_\alpha^i,\ 1 \leq i \leq N^A, \alpha \in \{0,1\}^{q_A}\right\}$ of the state space $\{0,1\}^{n_A}$ that is compatible with every SCC decompositions of the dynamics of modules Σ^A. By construction, the number of elements in this partition, denoted by M_A, verifies $1 \leq M_A \leq 2^{n_A}$. Applying the exact same procedure for module Σ^B, one obtains a similar partition $Z_h^B = \left\{\mathbf{B}_\beta^j,\ 1 \leq j \leq N^B, \beta \in \{0,1\}^{q_B}\right\}$ of the state space $\{0,1\}^{n_B}$, containing M_B elements.

Once partitions Z_h^A and Z_h^B are defined, the construction of the cross graph closely resembles the one of the asymptotic graph. The cross graph is the digraph $G^{cr} = (V^{cr}, E^{cr})$, where the vertex set V^{cr} is composed of all cross-products $\mathbf{A}_\alpha^i \times \mathbf{B}_\beta^j$ and the arc set is constructed as follows:

- $\mathbf{A}_\alpha^i \times \mathbf{B}_\beta^j \to \mathbf{A}_{\alpha'}^{i'} \times \mathbf{B}_\beta^j$ iff there exist $a \in \mathbf{A}_\alpha^i$, $a' \in \mathbf{A}_{\alpha'}^{i'}$ such that there is a transition from a to a' in graph $G^{A,\beta}$,
- $\mathbf{A}_\alpha^i \times \mathbf{B}_\beta^j \to \mathbf{A}_\alpha^i \times \mathbf{B}_{\beta'}^{j'}$ iff there exist $b \in \mathbf{B}_\beta^j$, $b' \in \mathbf{B}_{\beta'}^{j'}$ such that there is a transition from b to b' in graph $G^{B,\alpha}$.

There is also a matricial form for the definition of G^{cr}. First, project each transition graph $G^{A,u}$ onto Z_h^A, leading to 2^{p_A} graphs, represented by their $M_A \times M_A$ adjacency matrices $H^{A,u}$, $u \in \{0,1\}^{p_A}$. These projections can be rather straightforwardly achieved since Z_h^A is a fragmentation of the SCC decomposition of $G^{A,u}$. Second, for each $\alpha \in \{0,1\}^{q_A}$, introduce the $M_A \times M_A$ diagonal matrix Δ_α^A such that entry $\left[\Delta_\alpha^A\right]_{ii} = 1$ if the output of the i-th element of Z_h^A is equal to α and 0 otherwise. Once similar objects $H^{B,\upsilon}$ and Δ_β^B have been constructed for module Σ^B, the cross-graph is simply defined by a generalization of formula (2):

$$G^{cr} := \bigvee_{(\alpha,\beta)\in\{0,1\}^{q_A}\times\{0,1\}^{q_B}} \left(H^{A,\beta} \otimes \Delta^{B,\beta} \vee \Delta^{A,\alpha} \otimes H^{B,\alpha}\right). \quad (3)$$

EXAMPLE 4. To illustrate this definition, let us consider two 2-dimensional, single-input single-output modules Σ^A and Σ^B, defined by their transition graphs given in **Figure 1A** and their output functions $h^A(x) = x_2$, $h^B(y) = y_1$. The full transition graph of the interconnection, built from (2), is depicted in **Figure 1B** and the cross-graph is depicted in **Figure 1C**: it is constructed from the two partitions $Z_h^A = \{\{00, 10\}, \{01, 11\}\} = \{\{*0\}, \{*1\}\}$ and $Z_h^B = \{\{00\}, \{10\}, \{01\}, \{11\}\}$. □

The interest of the cross-graph lies in the following theorem, establishing the one-to-one correspondence between the terminal SCCs of G^{cr} and the attractors of the interconnected network.

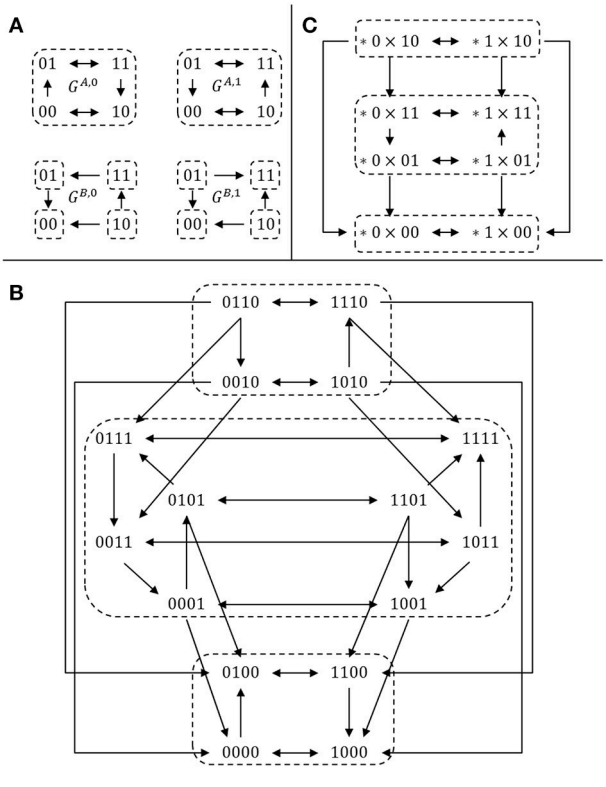


FIGURE 1 | Comparison between the cross graph of an interconnection and the full transition graph. **(A)** Transition graphs of two SISO modules (see Example 4); **(B)** full transition graph G of the interconnection; **(C)** cross graph G^{cr} of the interconnection. For each graph, dotted regions denote strongly connected components. There is a bijection between the SCC decomposition of the two graphs G (16 vertices) and G^{cr} (8 vertices), illustrating Theorem 2.

THEOREM 2. *Graphs G and G^{cr} have the same decomposition into strongly connected components. Furthermore, terminal SCCs of G^{cr} fully recover the attractors of the interconnected network.*

A proof of Theorem 2 is given in appendix. The size of the cross-graph is $M_A \times M_B$, which by construction is always less or equal than $2^{n_A+n_B}$, the size of the full interconnected graph G. The difference in size between the two graphs may vary greatly, and strongly depends on (i) the SCC decompositions of the two modules and (ii) as for the asymptotic graph, the numbers of inputs and outputs (and therefore the general modularity of the initial network). Part 4 proposes a brief evaluation of the performance of the method for a set of randomly generated interconnections. Although the interest of the cross-graph is mainly theoretical, in certain practical cases the full graph G can be too big to be stored easily while G^{cr} could.

Two possible extensions of the cross-graph method are noted here. First, Bérenguier et al. (2013) proposed a compression of the SCC graph of a network, called the hierarchical transition graph (HTG). As the cross-graph is constructed from a combination of the modules' SCC decompositions, it would be possible to consider similarly a combination of the modules' HTG decompositions. Benefiting from the compactness of HTGs, such a construction would be even more compact than the cross-graph. Second, note that both the cross graph and the asymptotic graph methods require prior analysis of the modules' dynamics and the computation of their attractors, implicitly implying the dimensions of the modules are manageable. For a large network, Zañudo and Albert (2013) proposed an efficient characterization of attractors with the notion of "stable motifs," based on the network's interaction graph (see also Klarner et al., 2015). When considering interconnections of large modules, the investigation of the stable motifs of an interconnection would therefore constitute an interesting extension of Theorem 2.

3.2. A Probabilistic Asymptotic Graph

One of the limitations of Boolean models is the lack of quantitative details: while the state transition graph describes all possible dynamical behaviors, it gives no indication as to which trajectory is more likely to be observed under a given set of initial conditions. To circumvent this problem, Boolean models can be combined with probabilistic frameworks that account for biological perturbations and variability in the logical rules (Shmulevich et al., 2002; Mori et al., 2015). Another approach is to exploit the Markov chain description of the transition graph associated to the asynchronous Boolean model (Calzone et al., 2010; Stoll et al., 2017). Based on this description, Stoll et al. (2017) developed the MaBoSS software, which then applies Gillespie algorithm to produce continuous time trajectories.

We also use the Markov chain description to assign probabilities to the edges of the asymptotic graph, an approach which will lead to a more quantitative analysis of the interconnected network's dynamics. The output of our probabilistic asymptotic graph is thus the set of attractors of the full network, under a particular interconnection scheme, together with a *relative probability* for each of them (e.g., "there is a probability p_1 that phenotype Q_1 is the outcome of this experiment").

The originality of our approach consists in assigning *incidence probabilities to the attractors of each separate module*, which can be obtained through the biological observations and measurements available for each module. The goal is to include biological information as an input and provide predictions that can be confronted to biological observations and therefore lead to validate or disprove the given interconnecting scheme.

3.2.1. Initializing Incidence Probabilities

Each transition in the asymptotic graph depends on two factors: which module is first "updated" (A or B) and, in response to an input change, how frequently does a switch occur from $A^i_{u\alpha}$ to $A^k_{\tilde{u}\tilde{\alpha}}$ (or from $B^j_{\upsilon\beta}$ to $B^k_{\tilde{\upsilon}\tilde{\beta}}$). These quantities may be represented by probabilities, defined *a priori*, from known data, experimental observations, or other modeling considerations.

Define

$$\varrho_A = P(\text{updating module } A \text{ first}).$$

Assume Boolean module Σ^A has a total of L_A same-output attractor-sets and Σ^B a total of L_B same-output attractor-sets,

$$\{A^{[i]} : A^i_{u_i\alpha_i},\ i = 1, \ldots, L_A\},\ \ \{B^{[j]} : B^j_{\upsilon_j\beta_j},\ j = 1, \ldots, L_B\},$$

and each of these has a given *incidence probability* (meaning that it is observed with a certain frequency) defined as

$$P(A^{[i]}) = w^i_A,\ \ i = 1 \ldots L_A, \quad P(B^{[j]}) = w^j_B, \quad j = 1 \ldots L_B.$$

The probabilities w^i_A and w^j_B may be assigned in different ways, for instance using experimental observations, or setting uniform probabilities ($w^i_A = 1/L_A$ for all i), or else from the size of their respective basin of attraction

$$w^i_A = \frac{\#\text{basin}^i}{\sum_i \#\text{basin}^i}, \tag{4}$$

but in any case they should satisfy $\sum_{i=1}^{L_A} w^i_A = 1$. Using these initial probabilities, a *joint incidence probability* may similarly be defined for each product of attractor-sets:

$$P(A^{[i]} \times B^{[j]}) = w^i_A w^j_B, \quad \Rightarrow \quad \sum_{i=1}^{L_A}\sum_{j=1}^{L_B} P(A^{[i]} \times B^{[j]}) = 1.$$

3.2.2. Transition Probabilities in the Asymptotic Graph

The probability of switching between two attractor-sets of the same module, but different inputs, can be defined in terms of conditional probabilities: define s^{ik}_A to be the probability that attractor $A^{[k]}$ is reached, conditional to the fact that the initial state is some $a^i \in A^{[i]}$. In other words, w^k_A must be weighted by the probability of a^i reaching any attractor in G^{A,u_k}:

$$s^{ik}_A = P(A^{[k]} | [a^i \in A^{[i]}]) = \frac{P(A^{[k]})}{\sum_{j\in\mathcal{J}} P(A^{[j]})} = \frac{w^k_A}{\sum_{j\in\mathcal{J}} w^j_A}, \tag{5}$$

where $\mathcal{J} = \{j : u_j = u_k \text{ and } a^i \rightsquigarrow A^{[j]}\}$ means that there exists a path in G^{A,u_k} leading from a^i to $A^{[j]}$, where $A^{[j]}$ is an attractor of G^{A,u_k}. A similar definition holds for s_B^{ik}.

Next, we can define the probability associated to an edge of V^{as} as:

$$\begin{aligned} P(A^{[i]} \times B^{[j]} \to A^{[k]} \times B^{[j]}) &= \bar{\varrho}_A s_A^{ik}, \\ P(A^{[i]} \times B^{[j]} \to A^{[i]} \times B^{[k]}) &= (1 - \bar{\varrho}_A) s_B^{jk}, \end{aligned} \tag{6}$$

with an "effective" probability $\bar{\varrho}_A$, computed based on the set of all ougoing edges from node $A^{[i]} \times B^{[j]}$:

$$\bar{\varrho}_A = \begin{cases} 0, & A^{[i]} \equiv A^{[k]} \\ 1, & B^{[j]} \equiv B^{[k]} \\ \varrho_A, & \text{otherwise.} \end{cases} \tag{7}$$

In other words, $\bar{\varrho}_A = 0$ if all outgoing edges have a fixed A-attractor, $A^{[i]} \times B^{[j]} \to A^{[i]} \times B^{[k]}$; $\bar{\varrho}_A = 1$, if all outgoing edges have a fixed B-attractor $A^{[i]} \times B^{[j]} \to A^{[k]} \times B^{[j]}$; $\bar{\varrho}_A = \varrho_A$ if outgoing edges may be of both types.

Note that these definitions ensure that the probabilistic asymptotic graph matrix has the property that all rows add up to 1:

$$\begin{aligned} &\sum_k P(A^{[i]} \times B^{[j]} \to A^{[k]} \times B^{[j]}) \\ &\quad + \sum_k P(A^{[i]} \times B^{[j]} \to A^{[i]} \times B^{[k]}) \\ &\quad = \sum_k \bar{\varrho}_A s_A^{ik} + \sum_k (1 - \bar{\varrho}_A) s_B^{jk} = \bar{\varrho}_A + (1 - \bar{\varrho}_A) = 1 \end{aligned}$$

since both $\sum_k s_A^{ik} = 1$ and $\sum_k s_B^{jk} = 1$.

3.2.3. Relative Probabilities of the Attractors of an Interconnection

If the asymptotic graph G^{as} has two or more attractors, in addition to the transition probabilities, another useful information is the frequency of observing a given attractor, or in other words the *relative probability of each attractor* of the interconnection. This probability can be computed from the SCC graph $G^{Sd} = (V^{Sd}, E^{Sd})$ corresponding to G^{as}, which is an acyclic graph and can be represented by an absorbing Markov chain. By definition, V^{Sd} is composed of the strongly connected components of G^{as}. Let $C \in V^{Sd}$ contain L_C elements of V^{as}. Define the incidence probability of observing C as:

$$P(C) = \sum_{\ell=1}^{L_C} P(A^{[i(\ell)]} \times B^{[j(\ell)]}) = \sum_{\ell=1}^{L_C} w_A^{i(\ell)} w_B^{j(\ell)}.$$

Moreover, a probability of transition can also be associated to each edge of E^{Sd}, $P(C^i \to C^j)$, computed by adding all the probabilities of the edges in E^{as} that link elements of C^i to elements of C^j. Suppose there are m strongly connected components, $|V^{Sd}| = m$, and let the $m \times m$ matrix M with $M_{ij} = P(C^i \to C^j)$, be the absorbing Markov chain associated with the graph G^{Sd}. Suppose M has r absorbing states, $\{C_a^k : k = 1, \ldots, r\}$, these are also the attractors of G^{Sd}. Matrix M can be written in the following canonical form (Feller, 1970):

$$M = \begin{bmatrix} Q & R \\ 0 & I_r \end{bmatrix},$$

where I_r is the $r \times r$ identity matrix, Q is the $(m-r) \times (m-r)$ matrix of transitions between transient states and R is the $(m-r) \times r$ matrix of transitions from transient states to absorbing states. Since M is irreducible, it follows that $(I-Q)$ has an inverse (where I is the $(m-r) \times (m-r)$ identity matrix). Then the probability that there exists a path from a given state to one of the r absorbing states is given by the probability of being absorbed by r:

$$M_{absorp} = (I - Q)^{-1} R,$$

where $M_{absorp}(i,k)$ is the probability that transient state i converges to absorbing state k.

If, in addition, we wish to weigh these absorption probabilities by the incidence probabilities of observing C_a^k, we can define the *relative probability of an attractor of the asymptotic graph*:

$$P_{rel}(C_a^k) = P(C_a^k) + \sum_{i=1}^{m-r} M_{absorp}(i,k) P(C^i), \quad k = 1, \ldots, r \tag{8}$$

where C_a^k denotes each attractor and $P(C_a^k)$ is the incidence probability of C_a^k.

4. PERFORMANCE ON RANDOM NETWORKS' INTERCONNECTIONS

In this part we propose a series of computational experiments to assess the efficiency of the asymptotic graph and the cross graph to recover the attractors of random interconnected Boolean networks. Following the general idea of inputs/outputs at the core of this paper, we start with a brief description of the algorithm used to generate random IO modules. We then present numerical results computed on random interconnections with varying connectivity, showing the respective advantages and limitations of the two methods in practice.

4.1. Generation of Random IO Networks With Varying Connectivity

The NK-model, introduced by Kauffman (1969), is a general statistical model to represent random Boolean networks by controlling their dimension N and their inner connectivity K. It is used for instance by Zañudo and Albert (2013) and Veliz-Cuba et al. (2014). Here it is slightly adapted to include inputs and outputs. Let Σ be an IO Boolean network of dimension (n, p, q), of transition function $f : \{0,1\}^p \times \{0,1\}^n \to \{0,1\}^n$ and output function $h : \{0,1\}^n \to \{0,1\}^q$. A usual way to depict such a network is by its wiring diagram, showing the dependencies between the different variables of the network. Equivalently, the wiring diagram can be represented by a $(n+q) \times (p+n)$ Boolean matrix

$$M = \left[\begin{array}{c|c} A & B \\ \hline 0 & D \end{array}\right],$$

where submatrices A $(n \times p)$, B $(n \times n)$ and D $(q \times n)$ are defined as follows:

$$a_{ij} = \begin{cases} 1 \text{ if function } f_i \text{ depends explicitly of input variable } u_j, \\ 0 \text{ otherwise,} \end{cases}$$

$$b_{ij} = \begin{cases} 1 \text{ if function } f_i \text{ depends explicitly of variable } x_j, \\ 0 \text{ otherwise,} \end{cases}$$

$$d_{ij} = \begin{cases} 1 \text{ if output function } h_i \text{ depends explicitly of variable } x_j, \\ 0 \text{ otherwise.} \end{cases}$$

Let C designate the matrix $[A|B]$. The sum of the i-th row of C is the number of essential variables of logical function f_i, also called the *connectivity* of f_i. Given integers $n > 0$, $p, q \geq 0$ and a real number $K_{mean} \in [1, n]$, we construct a random IO network of dimension (n, p, q) and of average connectivity K_{mean} by applying the following procedure, which generates a dependency matrix M:

1. Let $D := 0$. For each $1 \leq i \leq q$, pick at random $j \in \{1, \ldots, n\}$ and set $d_{ij} := 1$.
2. Generate n integers k_i in $\{0, \ldots, n+p\}$ according to a binomial distribution of parameters $n + p$ (number of trials) and $\frac{K_{mean}}{n+p}$ (probability of success).
3. Let $C = [A|B] := 0$. For each $1 \leq i \leq n$, pick a random combination $(j_1, \ldots, j_{k_i}) \in \{1, \ldots, n + p\}^{k_i}$ (without replacement) and set $c_{i,j_l} := 1$ for all $1 \leq l \leq k_i$.
4. Check that each column of A is non-zero; while it is not the case, repeat step 3.
5. Set $M := \left[\begin{array}{c} C \\ \hline 0 \,|\, D \end{array}\right]$.

Step 4 ensures the generated module actually depends of every inputs. Once the dependency matrix M is obtained, the last step consists in generating the $n + q$ Boolean functions according to M. A Boolean function of k variables is picked randomly among the 2^{2^k} possibilities; in case it is degenerate (i.e., at least one of the k variables is not essential), another one is chosen so as to ensure exact compatibility with M.

4.2. Complementarity of the Cross and Asymptotic Graph Methods

With this algorithm, it is possible to generate a IO module by controlling its inner connectivity, that is the number of actual dependencies in the wiring diagram. Thus, it becomes possible to generate random interconnections with varying degrees of *modularity*, according to the average connectivity of each module. We used this algorithm to generate 2,000 interconnections of two modules Σ^A and Σ^B of dimensions $(n_A, p_A, q_A) = (n_B, p_B, q_B) = (10, 2, 2)$:

$$\boxed{\Sigma^A} \rightleftarrows \boxed{\Sigma^B}, \tag{9}$$

where the mean connectivity of Σ^A and Σ^B varies in $\{1, \ldots, 10\}$. For each interconnection, both 10-dimensional modules were analyzed separately (including the computation of the transition graphs, their SCC decompositions and the computation of their attractors), then the cross graph and the asymptotic graph were computed and compared. The main results are presented in **Figure 2** and summarized below. All computations were made with Matlab R2016b, The MathWorks, Inc.

First, we compare the respective sizes N^{cr} and N^{as} of the cross and the asymptotic graphs (*ie.* their number of vertices). **Figures 2A,B** show respectively the evolution of $\log_2(N^{cr})$ and $\log_2(N^{as})$ with respect to the connectivity of the two modules. Obviously, both N^{cr} and N^{as} are below $N = 2^{20}$, which is the size of the full transition graph of the interconnected network. The cross graph, which captures both the transient and the asymptotic dynamics of the interconnection is relatively large, however its size seems to vary greatly with the modules' connectivity. When the connectivity increases, implying a highly modular interconnection, the ratio N^{cr}/N can reach very small values, emphasizing the interest of the cross graph to efficiently store the dynamics of large, modular interconnected networks. On the other hand, the asymptotic graph is always much smaller, several orders of magnitude under the size of the full transition graph. Contrary to the cross graph, it is particularly small when the modules have lower connectivity, making it particularly well adapted for biological networks. Interestingly, its size seems to reach a plateau when the mean connectivity is above $\frac{n}{2} = 5$.

Another way to compare the two approaches is by studying their average execution times. The times shown in **Figure 2C** include the analysis of the two 10-d modules and of the cross and asymptotic graph methods. The latter comprise the construction of G^{cr} (respectively, of G^{as}), the SCC decomposition of G^{cr} (respectively, of G^{as}) and the reconstruction of the attractors (respectively, of $\pi(R)$ for all terminal SCCs R of G^{as}). For the cross graph, the majority of the time is taken by the SCC decomposition of G^{cr} while for the asymptotic graph, the most time-consuming step is the construction of G^{as} itself (data not shown). For comparison, we also computed the complete dynamics of the 20-d interconnected network by using formula (2); on average, such direct method amounted to around 83 seconds (dotted line). Therefore, both methods are faster than the direct analysis of the full interconnected network. As before, the asymptotic graph is particularly efficient for low connectivity modules, while the cross graph is more efficient when the modules have high connectivity. Interestingly, for connectivity $K_{mean} = 5$ and higher, when both graphs have roughly the same size, the cross graph method becomes even more rapid than the asymptotic graph.

Finally, since both graphs were computed it was possible to evaluate the quality of the asymptotic graph predictions. Recall that according to Theorem 1, the asymptotic graph has two drawbacks. First, it may predict spurious attractors and second, when it identifies a true attractor it only predicts a subset $\pi(R)$ of the states lying in the attractor Q. The ratio $\frac{|\pi(R)|}{|Q|}$ is called the *accuracy* of the prediction. Among the 2,000 interconnections, 11 presented spurious attractors that is only 0.55% of the total. In all but one case, only one spurious attractor was detected. This result confirms the rarity of the appearance of spurious attractors. In total, we identified 3,693 true attractors. Among them more than 73% were completely recovered (see **Figure 2D**); overall, the mean accuracy is about 0.86, exhibiting the excellent predictive power of the asymptotic

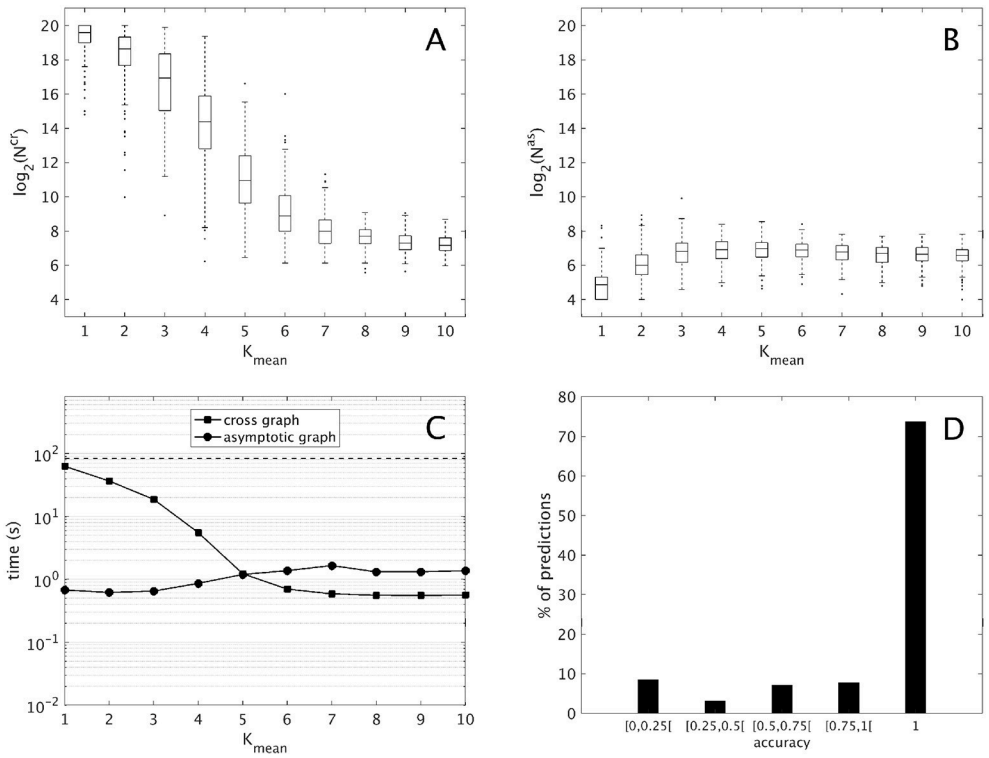


FIGURE 2 | Computational results for 2,000 interconnections of two 10-dimensional modules [according to (9)], split into ten groups of 200 sorted along K_{mean}, the mean connectivity of the modules. **(A,B)** Evolution of the sizes of the cross graph and of the asymptotic graph, $\log_2(N^{cr})$ and $\log_2(N^{as})$ with respect to the modules' connectivity (obtained with the routine `boxplot` of Matlab's Statistics toolbox). **(C)** Mean execution time in seconds of the cross and asymptotic graph methods (logarithmic scale). The dotted line represents the average time of the direct method (analysis of the full interconnected network). **(D)** Histogram of the accuracies of all the attractors predicted by the asymptotic graph (3,693 attractors in total).

graph when it comes to uncover the asymptotic behaviors of an interconnection.

4.3. A Powerful Tool to Analyze Large Interconnections of Biological Networks

According to the previous results, the asymptotic graph seems particularly well adapted when the mean connectivity of the modules is low ($\leq$ 5), which is arguably where biological networks generally operate (Zañudo and Albert, 2013; Veliz-Cuba et al., 2014). Therefore we decided to test it further with higher dimensional interconnections, including four modules $\Sigma^A, \Sigma^B, \Sigma^C, \Sigma^D$ of dimension $n = 15$, with $K_{mean} \in \{1, \ldots, 5\}$, $p_A = q_A = p_D = q_D = 1$ and $p_B = q_B = p_C = q_C = 2$:

$$\boxed{\Sigma^A} \underset{\leftarrow}{\rightarrow} \boxed{\Sigma^B} \underset{\leftarrow}{\rightarrow} \boxed{\Sigma^C} \underset{\leftarrow}{\rightarrow} \boxed{\Sigma^D} \tag{10}$$

When $N^{cr} < 10^7$, the cross graph was also constructed and analyzed, in order to check the existence of spurious attractors. Since the global state space is $2^{60} > 10^{18}$, we skipped the last treatment (identification of the attractors in $\{0,1\}^{60}$) to avoid possible explosions. Therefore, we only computed the terminal SCCs of G^{as} and, when available, the terminal SCCs of G^{cr}. The results are presented in **Table 1**. When G^{cr} could be analyzed, we were able to detect spurious attractors in G^{as}: none were found. If the cross graph method is not practical for small K_{mean}, the asymptotic graph was always manageable, confirming its practical interest to analyze large biological networks, as long as they can be expressed as interconnections of modules with a reasonable number of inputs and outputs.

5. TWO BIOLOGICAL APPLICATIONS

The asymptotic graph construction and its probabilistic interpretation are now applied to two biological examples, centered on the mammalian and yeast cell cycles. Both cases illustrate the asymptotic graph concept, its informative description of a composite system, and its usefulness for testing biological hypotheses.

5.1. Mammalian Cell Cycle, Circadian Clock and Their Interconnection

There are two basic cellular oscillators in mammalian cells: cell cycle describes the different phases of cellular growth and division, while circadian clock decribes the mechanism responsible for anticipating environmental changes and adapting the organism to deal with these changes (most notably, day-night differences). The interactions between these two oscillators are

TABLE 1 | Computational results for 200 interconnections of four 15-dimensional modules [according to (10)], split into five groups of 40, sorted along the mean connectivity of the modules.

K_{mean}	$\log_2(N^{cr})$		Time (s)			$\log_2(N^{as})$		Time (s)		
	mean	std	mean	std	#treated/(#exp.)	mean	std	mean	std	#spurious/(#treated)
1	57.3	2.3	–	–	0/40	8.5	1.4	9	2	–
2	52.2	4.1	–	–	0/40	9.8	1.1	9	7	–
3	42.4	5.7	–	–	0/40	11.0	1.4	63	185	–
4	29.6	5.9	493	361	6/40	11.3	1.1	40	51	0/6
5	20.9	4.7	176	223	28/40	11.0	1.0	27	38	0/28

The cross graph is treated (constructed and analyzed) only when $N^{cr} < 10^7$ ($\log_2(10^7) \approx 23.25$). The column #treated/(#exp.) indicates the number of times it was treated over the total number of experiments. When it is treated, we further verify the presence of spurious attractors in the asymptotic graph. The column #spurious/(#treated) indicates the number of times the asymptotic graph predicts a spurious attractor over the number of times the cross graph could be treated. Symbol — indicates that the corresponding value could not be computed.

still not fully understood, but recent works by Feillet et al. (2014) and Bieler et al. (2014) have uncovered unexpected bi-directional links between the two modules. Successful mathematical models for the cell cycle and clock have been developed, as well as some studies on their interactions (Gérard and Goldbeter, 2012), but many questions remain (Feillet et al., 2015).

5.1.1. Mammalian Boolean Modules

At the discrete level, a reference model of the cell cycle was developed and discussed by Fauré et al. (2006) (see **Figure 3**). It comprises 10 variables:

$$(CycD, Rb, E2F, CycE, CycA, p27, Cdc20, Cdh1, Ubc, CycB),$$

where $CycX$ ($X \in \{A, B, D, E\}$) represent four cyclins, each roughly corresponding to one of the four phases of the cell cycle. This constitutes our module Σ^A, and its rules can be found in the Supplementary Material. The clock model (module Σ^B) has 7 variables and is based on the work of Comet et al. (2012). To account for transcription shutdown during mitosis, the input υ negatively affects all mRNAs:

$$\begin{aligned} BMAL^+ &= \neg PCnuc \\ mPER^+ &= \neg\upsilon \wedge BMAL \\ mCRY^+ &= \neg\upsilon \wedge BMAL \\ pPER^+ &= mPER \\ pCRY^+ &= mCRY \\ PC^+ &= pPER \wedge pCRY \\ PCnuc^+ &= PC. \end{aligned} \tag{11}$$

In the clock model, mX and pX denote mRNA and protein coded by gene X, while PC denotes the complex formed by the proteins PER and CRY, and $PCnuc$ denotes this complex in the nucleus.

A well established link between these two oscillators is that protein $BMAL$ acts on the cell cycle, possibly at different stages (Feillet et al., 2015). In our analysis, we will consider $BMAL$ acting during G1 phase. Although no conclusive evidence exists on how the cell cycle may affect the clock, we have considered that during cell division (or mitosis phase) gene expression is stopped (in the model, mitosis can be modeled as $Cdc20 \wedge CycB$,

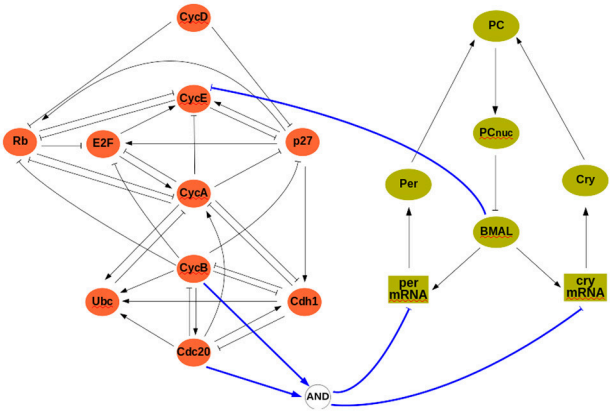


FIGURE 3 | The interconnected mammalian cell cycle (**Left**, adapted from Fauré et al., 2006) and clock (**Right**, adapted from Comet et al., 2012). Square symbols represent messenger RNAs. Solid blue arrows denote input/output connections.

see **Figure 3**). The interconnection between modules is thus given by:

$$u = h^B(b) = BMAL, \qquad \upsilon = h^A(a) = Cdc20 \wedge CycB,$$

so that $u = 0$ (resp., $u = 1$) represents absence (resp., presence) of $BMAL$ and $\upsilon = 1$ represents mitosis. In the cell cycle model, $BMAL$ affects negatively the G1 phase, leading to a logical equation for cyclin E of the form $cycE^+ = \neg u \wedge (E2F \wedge \neg Rb)$ (see **Figure 3** and Supplementary Material).

Module Σ^A has a total of six, and module Σ^B has a total of three, same-output attractor sets. For algorithmic convenience, these are labeled using the lexicographic convention, that is $A^j_{\hat{u}\hat{\alpha}}$ for $\hat{u}, \hat{\alpha} \in \{1, 2\}$, where "decimal 1 = logical 0" and "decimal 2 = logical 1." The attractors for both modules are as follows:

$$\begin{aligned} G^{A,u=0}&: \ A^1_{11} = \{0100010100\},\ A^2_{11}(80 \text{ states}),\ A^3_{12}(32 \text{ states}), \\ G^{A,u=1}&: \ A^4_{21} = \{0100010100\},\ A^5_{21}(40 \text{ states}),\ A^6_{22}(16 \text{ states}), \\ G^{B,\upsilon=0}&: \ B^1_{11}(57 \text{ states}),\ B^2_{12}(63 \text{ states}), \\ G^{B,\upsilon=1}&: \ B^3_{22} = \{1000000\}. \end{aligned}$$

In the case $u = 0$, module Σ^A becomes exactly the original model constructed by Fauré et al. (2006). Therefore, as expected, the attractors found for $G^{A,u=0}$ correspond exactly to those listed by Fauré et al. (2006). Attractors A^1_{11} and A^4_{21} correspond to a steady state where the only expressed proteins are *Rb*, *p*27, and *Cdh*1, hence representing the quiescent cell state. The (full) attractor $A^2_{11} \cup A^3_{12}$ is a cyclic attractor containing 112 distinct states and corresponds to the known G1/S/G2/M cell cycle progression (Fauré et al., 2006). Similarly, $A^5_{21} \cup A^6_{22}$ is a cyclic attractor of the graph $G^{A,u=1}$, with 56 states. It tends to describe the cell cycle progression, with the difference that $u = 1$ implies $CycE \equiv 0$. In either of the cyclic attractors, the attractor-sets A^3_{12} and A^6_{22} contain states representing mitosis, that is, the output of any state $a \in A^6_{22} \cup A^3_{12}$ satisfies $h^A(a) = Cdc20 \wedge CycB = 1$.

The clock mechanism admits a cyclic attractor with 120 states, $B^1_{11} \cup B^2_{12}$, which corresponds to regular circadian oscillations in the case $\upsilon = 0$. At mitosis, represented by $\upsilon = 1$, the clock network admits a single steady state attractor ($B^3_{22} = \{1000000\}$), where all gene expression is arrested.

5.1.2. Asymptotic and Cross Graphs

The asymptotic graph for the interconnection of the two mammalian oscillators has 18 nodes and two attractors, with separate basins of attraction (**Figure 4**):

$$\begin{aligned} R_1 &= \{A^1_{11} \times (B^1_{11} \cup B^2_{12}), A^4_{21} \times (B^1_{11} \cup B^2_{12})\} \\ R_2 &= \{(A^2_{11} \cup A^3_{12}) \times B^1_{11}, A^2_{11} \times B^2_{12}, A^5_{21} \times (B^1_{11} \cup B^2_{12}), \\ &\quad (A^5_{21} \cup A^6_{22}) \times B^3_{22}, A^6_{22} \times B^2_{12}, A^3_{12} \times B^3_{22}\}. \end{aligned}$$

The cross graph contains 54,272 nodes (compare to the full size of the interconnection, $2^{17} = 131072$) and confirms the existence of exactly two cyclic attractors for the interconnected system and returns all their elements: attractor R_1 is composed of 120 states and R_2 is composed of 13,552 states.

Our methodology predicts two distinct operating modes for the coupled oscillators: R_1 corresponds to a quiescent cell with oscillatory clock, since it is the product of state 0100010100 representing a quiescent cell in module Σ^A and of cyclic attractor $B^1_{11} \cup B^2_{12}$ representing regular clock oscillations. The attractor R_1 is thus in agreement with observations by Plikus et al. (2013) (hair cells in quiescent phase seem to have a running clock). In contrast, R_2 represents joint oscillations of the cell progression cycle ($A^2_{11} \cup A^3_{12}$) and clock ($B^1_{11} \cup B^2_{12}$) (see **Figure 4** for the dynamics within R_2). The cell cycle and clock may jointly oscillate and alternate states with a regular cycle of cyclin E (which is present mostly through S phase and mitosis) or eventually switch to a joint cycle with absence of cyclin E ($A^5_{21} \times B^1_{1.} \rightarrow A^2_{1.} \times B^1_{11} \rightarrow A^2_{11} \times B^2_{12} \rightarrow A^5_{21} \times B^1_{1.}$). However, at mitosis ($A^3_{12}$), the clock may switch to its arrested steady state ($A^3_{12} \times B^1_{11} \rightarrow A^3_{12} \times B^3_{22}$), which leads directly to a full degradation of cyclin E in the cell cycle ($A^5_{2.} \times B^3_{22}$).

To assign transition probabilities to the asymptotic graph, there are essentially two elements to define: ϱ_A which is the probability of updating first the component from module Σ^A; and the incidence probability of each attractor from each module, w^i_A and w^j_B. To compute the incidence probabilities w^i_A and w^j_B, we have used the size of the original basins of attraction of $A^i_{u\alpha}$ in Σ^A and $B^j_{\upsilon\beta}$ in Σ^B, as in (4). However, for both modules, each attractor can be reached from any state, implying that the joint incidence probabilities, $P(A^{[i]} \times B^{[j]}) = w^i_A \times w^j_B$, are equal for all nodes of the asymptotic graph with: $w^i_A = 1/6$ ($i = 1, \ldots 6$) and $w^j_B = 1/3$ ($j = 1, \ldots 3$).

Figure 4 shows the transition probabilities obtained for two different values of the updating probability ϱ_A. These two graphs are very similar, differing only on the most frequent transitions (bold arrows, above 0.5). As should be expected, whenever the probability of first updating components from Σ^A is larger ($\varrho_A = 0.6$), the cell cycle oscillations dominate the global dynamics: most of the bold transitions in **Figure 4** (bottom) concern switches between attractor-sets of Σ^A. In contrast, circadian clock oscillations are dominant for $\varrho_A = 0.2$ (**Figure 4**, top). The evolution from mitosis phase toward cell cycle progression ($A^3_{12} \times B^3_{22} \rightarrow A^5_{21} \times B^3_{22}$ or $A^3_{12} \times B^3_{22} \rightarrow A^6_{22} \times B^3_{22}$) is equally probable for either ϱ_A.

Computation of the relative probabilities (8) of reaching one of the attractors of the interconnected network yields

$$P_{rel}(R_1) = 0.333, \quad P_{rel}(R_2) = 0.667,$$

independently of the updating probability ϱ_A. An interpretation of these relative probabilities is that, in a typical population of cells, about one third are arrested in quiescent G0 state while the other two thirds follow the normal cell cycle progression G1/S/G2/M.

5.2. Budding Yeast Cell Growth and Cell Cycle START

Cell cycle and division is intimately linked with cell growth: a cell cannot divide into two daugther cells if its size is too small. There are many other factors that play a role in cell division (concentration of certain proteins, volume), but it remains unclear how a cell is able to perceive its own size and evaluate whether all conditions are in place for cell division (Turner et al., 2012).

In budding yeast, cell cycle is triggered by a START signal which is dependent on cell size. Li et al. (2004) propose a Boolean model that accurately describes cell cycle progression, taking START as an external input and stopping at a G1 phase steady state. One of the most important proteins involved in START is cyclin *Cln*3, which in involved in the G1-S phase transition and initiates cell cycle in the model of Li et al. (2004). Cyclin *Cln*3 forms a complex with another protein *Whi*3 but, in order to initiate cell cycle, *Cln*3 must be folded and released from this complex, which is achieved with the help of a chaperon protein *Ydj*1. Recent work by Aldea et al. (2017) suggests that cell size is growth rate dependent and that *Ydj*1 is one of the most important factors relating growth rate to cell size at START.

5.2.1. Budding Yeast Boolean Modules

A reference discrete model for the cell cycle was developed by Li et al. (2004). It comprises 11 variables:

$$(START, MBF, SBF, Cln1, Cdh1, Swi5, Cdc20, Clb5, Sic1, Clb1, Mcm)'$$

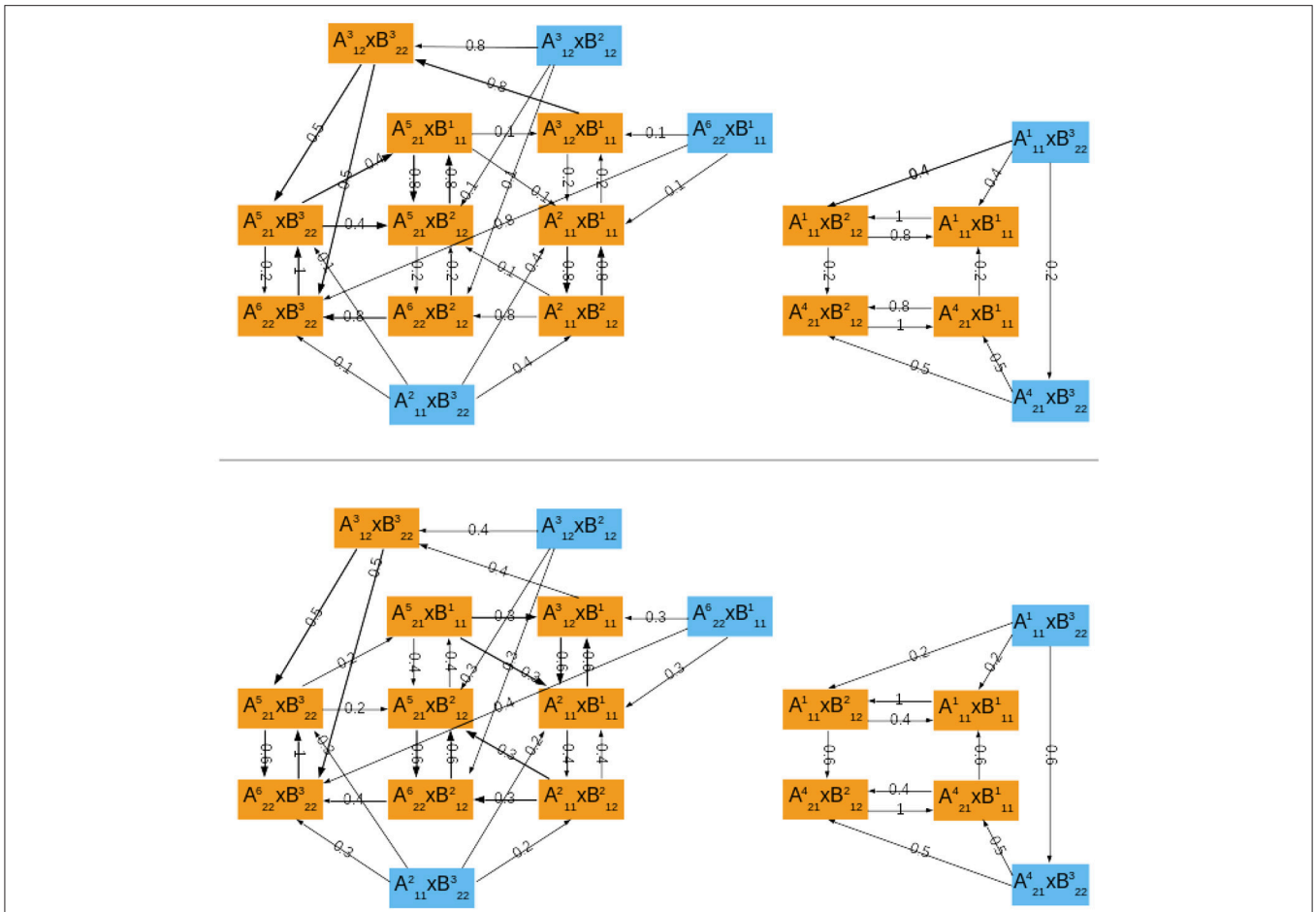


FIGURE 4 | The probabilistic asymptotic graph for the interconnected mammalian oscillators. Orange colored nodes belong to an attractor: R_1 at right and R_2 at left. Bold arrows represent transitions with probability ≥ 0.5. **(Top)** $\varrho_A = 0.2$. **(Bottom)** $\varrho_A = 0.6$.

with *START* given by *Cln*3 (see **Figure 5**; the Boolean rules can be found in the Supplementary Material).

To describe cell size dependence on growth rate Aldea et al. (2017) proposes a model where *Cln*3 competes with a second hypothetical protein *Prot* for binding with *Ydj*1 for folding:

$$\begin{aligned} Prot + Ydj1 &\leftrightharpoons YP \rightarrow ProtF + Ydj1 \\ Cln3 + Ydj1 &\leftrightharpoons YC \rightarrow Cln3F + Ydj1, \end{aligned}$$

and *Prot* would be a growth rate dependent protein. Here, we propose a basic Boolean network of this model, where the dependence on growth rate is modeled by an input υ:

$$\begin{aligned} Ydj1^+ &= YP \vee YC \vee \neg(Prot \wedge Cln3) \\ YP^+ &= Ydj1 \wedge Prot \\ YC^+ &= Ydj1 \wedge Cln3 \\ Prot^+ &= \upsilon \\ ProtF^+ &= YP \\ Cln3^+ &= \neg Whi3 \\ Cln3F^+ &= YC \\ Whi3^+ &= \upsilon. \end{aligned} \tag{12}$$

The competition of *Prot* and *Cln*3 for *Ydj*1 is represented by the term $\neg(Prot \wedge Cln3)$ in the rule for *Ydj*1 meaning that, in the absence of both *Prot* and *Cln*3, "free" protein *Ydj*1 will be available. Both *Prot* and *Whi*3 depend on growth rate, here given by input υ. Later on, υ will be computed as an output from the cell cycle model.

Computation of the graphs $G^{A,u}$ and $G^{B,\nu}$ yields the following attractors:

$$\begin{aligned} G^{A,u=0}: \quad & A_{11}^1 = \{00000000000\},\ A_{11}^2 = \{00000000100\}, \\ & A_{11}^3 = \{00001000000\}, \\ & A_{1*}^4 = \{00001000100\},\ A_{11}^5 = \{00110000000\}, \\ & A_{11}^6 = \{01000000100\}, \\ & A_{1*}^7 = \{01001000100\}, \\ G^{A,u=1}: \quad & A_{2*}^8 = \{10110110011\},\ A_{2*}^9 = \{11000111011\}, \\ & A_{2*}^{10} = \{11110110011\}, \\ & A_{2*}^{11} = \{11110111011\}, \end{aligned}$$

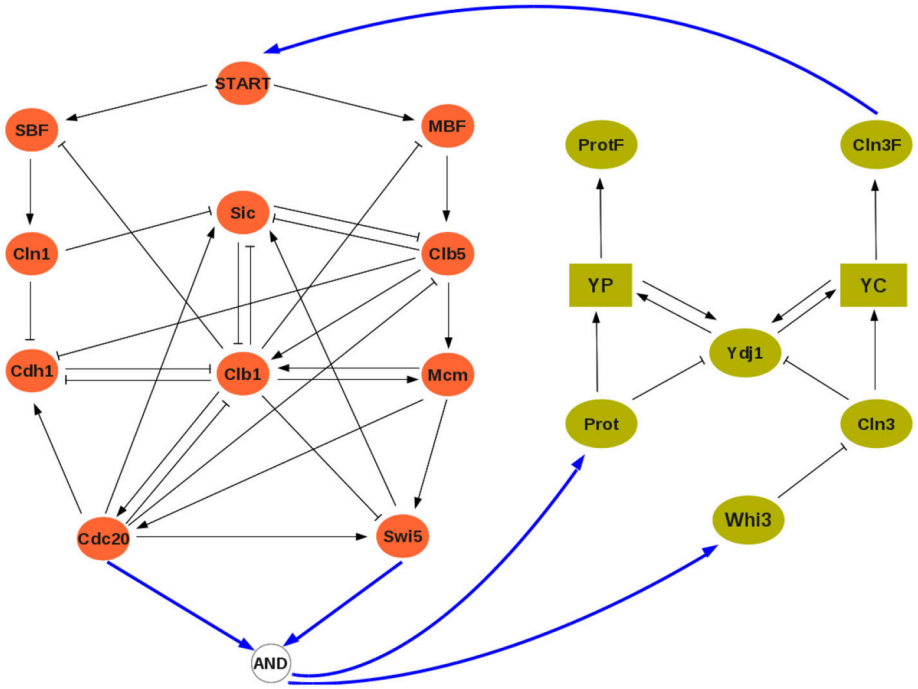


FIGURE 5 | The interconnected yeast cell cycle (**Left**, adapted from Li et al., 2004) and cell size network (**Right**, adapted from Aldea et al., 2017). Square symbols represent messenger RNAs. Solid blue arrows denote input/output connections.

$$G^{B,\upsilon=0}: \quad B^1_{12} = \{10100110\},$$
$$G^{B,\upsilon=1}: \quad B^2_{21} = \{11011001\}.$$

The symbol $*$ in A^i_{1*} or A^i_{2*} means that the output of this attractor depends on the function $h^A(a)$: three different forms for $h^A(a)$ will be tested (see 13–15 below). For instance, we have $h^A(A^4_{1*}) = 2$ whenever $h^A(a)$ is given by (15), so we should write A^4_{12}; but $h^A(A^4_{1*}) = 1$ in the other two cases, hence A^4_{11}.

In the case $u = 0$, the yeast cell cycle model is exactly the one studied by Li et al. (2004) hence, as expected, the seven attractors A^i_{1*} of $G^{A,u=0}$ are those listed in **Table 1** of this reference. According to Li et al. (2004), attractor A^4_{1*} represents the G1 steady state and has the largest attraction basin. Attractor A^2_{11} is also close to G1 phase and has the second largest attraction basin. Using the size of the attractions basins, the incidence probabilities w^i_A have been computed according to Equation (4) and they are listed in **Table 2**.

5.2.2. Network Interconnection, Asymptotic and Cross Graphs

To establish a scheme of interconnection, observe that the cell size model acts on the cell cycle by triggering the start signal, that is START is given by (folded/free) protein *Cln3F*. Conversely, the input of the cell cycle to the cell size module is still unknown, the combination of variables and/or quantities used by the cell to detect its own size is a question for further analysis. As an hypothesis, we will assume that growth rate is detected through cell phase, since the cell cycle model provides this information.

TABLE 2 | Interconnection of yeast models.

Attractor	Boolean representation	w^i_A
A^1_{11}	00000000000	0.0802
A^2_{11}	00000000100	0.0882
A^3_{11}	00001000000	0.0792
A^4_{1*}	00001000100	0.0893
A^5_{11}	00110000000	0.0669
A^6_{11}	01000000100	0.0472
A^7_{1*}	01001000100	0.0490
A^8_{2*}	10110110011	0.0921
A^9_{2*}	11000111011	0.1290
A^{10}_{2*}	11110110011	0.0749
A^{11}_{2*}	11110111011	0.2039
Attractor	**Boolean representation**	w^i_B
B^1_{12}	10100110	0.5
B^2_{21}	11011001	0.5

Attractors and incidence probabilities, proportional to the size of each basin of attraction. For module Σ_B, there exists a unique attractor for each v, hence $w^1_B = w^2_B$. The symbol $*$ means that the output of this attractor depends on the function $h^A(a)$: we have A^4_{12} with a G1 indicator but A^4_{11} in the other two cases.

To explore the plausibility of this hypothesis, we will thus consider three different indicators of the cell cycle phase (M, S, and G1 phases) and compare the asymptotic graphs of the three

corresponding interconnection schemes:

$$\text{M-phase}: u = h^B(b) = Cln3F, \quad v = h^A(a) = Swi5 \wedge Cdc20, \tag{13}$$

$$\text{S-phase}: u = h^B(b) = Cln3F, \quad v = h^A(a) = Clb5, \tag{14}$$

$$\text{G1-phase}: u = h^B(b) = Cln3F, \quad v = h^A(a) = Cdh1 \wedge Sic. \tag{15}$$

In the case of growth measured by M phase ($h^A(a) = Swi5 \wedge Cdc20$), the asymptotic graph has a unique, cyclic, attractor (**Figure 6**, top):

$$R_1^M = \{A_{22}^{11} \times B_{12}^1, A_{22}^{11} \times B_{21}^2, A_{11}^2 \times B_{21}^1, A_{11}^2 \times B_{12}^1, A_{11}^4 \times B_{21}^2, A_{11}^4 \times B_{12}^1\}$$

This information is confirmed and complemented by computation of the cross graph, which has 524,288 nodes ($= 2^{19}$). Attractor R_1^M is composed of 116,520 states.

Interestingly, although neither Σ^A nor Σ^B have periodic orbits, in this case the interconnected network does exhibit an oscillatory orbit: at stationary G1 (A_{11}^4) the START signal (B_{12}^1) is received and the module Σ^A performs one cell cycle:

$$A_{11}^4 \times B_{21}^2 \to A_{11}^4 \times B_{12}^1 \to A_{22}^{11} \times B_{12}^1 \to A_{22}^{11} \times B_{21}^2,$$

setting *Cln3* back to its OFF state (B_{21}^2) and ending "near" M phase (A_{22}^{11}). At this point, the system returns to stationary G1 and repeats the cycle, waiting for cell to grow and again send

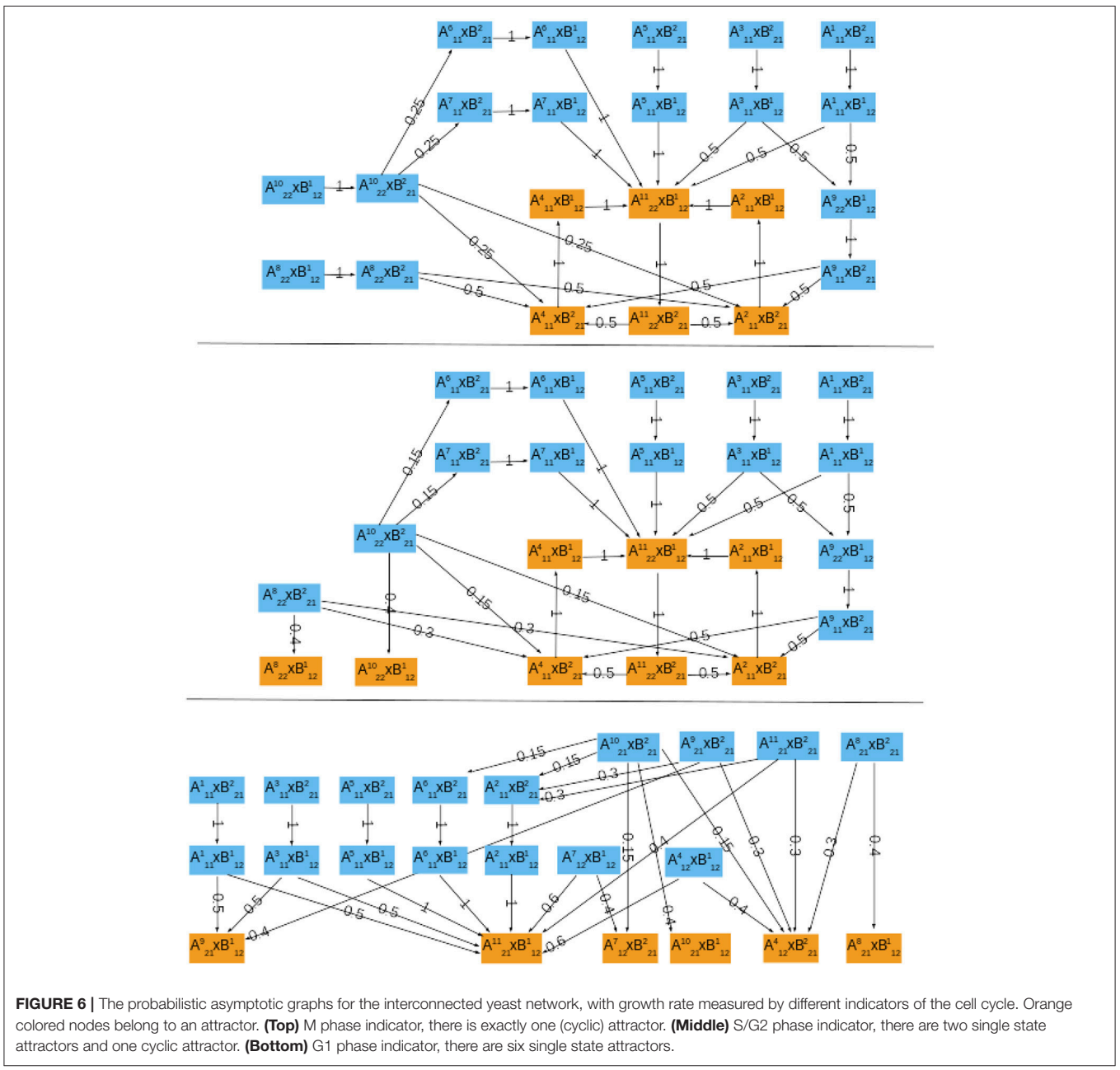


FIGURE 6 | The probabilistic asymptotic graphs for the interconnected yeast network, with growth rate measured by different indicators of the cell cycle. Orange colored nodes belong to an attractor. **(Top)** M phase indicator, there is exactly one (cyclic) attractor. **(Middle)** S/G2 phase indicator, there are two single state attractors and one cyclic attractor. **(Bottom)** G1 phase indicator, there are six single state attractors.

the start signal. Two alternative paths are proposed for the cell cycle, with G1-phase described either by A^4_{11} or similar state A^2_{11}. Since G^{as} contains a unique attractor, its relative probability P_{rel} is necessarily 1.

In the case of growth rate measured by S phase ($h^A(a) = Clb5$), the asymptotic graph (**Figure 6**, middle) has three attractors, two single steady state and one cyclic attractor:

$$\begin{aligned} R^S_1 &= \{A^{11}_{22} \times B^1_{12}, A^{11}_{22} \times B^2_{21}, A^2_{11} \times B^1_{21}, \\ &\quad A^2_{11} \times B^1_{12}, A^4_{11} \times B^2_{21}, A^4_{11} \times B^1_{12}\} \\ R^S_2 &= \{A^8_{21} \times B^1_{12}\} \\ R^S_3 &= \{A^{10}_{21} \times B^1_{12}\} \end{aligned}$$

In this case, however, computation of the cross graph shows that R^S_1 is a spurious attractor, implying that the asymptotic graph has lost some information on transient pathways. In practice, the full graph contains pathways eventually leading from R^S_1 to either R^S_2 or R^S_3. This example shows the importance of verifying whether any of the asymptotic graph's attractors is spurious, and hence the usefulness of a complementary method as the cross graph. In this situation, the probabilistic interpretation of the asymptotic graph is unclear. The relative probabilities computed according to (8) yield equal probabilities for reaching attractors R^S_2 and R^S_3 (see **Table 3**). In contrast, R^S_1 must now be interpreted as a transient set of states.

In the case G1 is used as measure of growth rate, we have $h^A(a) = Cdh1 \wedge Sic$ and the asymptotic graph (**Figure 6**, bottom) has six single state attractors but *no cyclic attractor*:

$$\begin{aligned} R^{G1}_1 &= A^4_{12} \times B^2_{21}, \; R^{G1}_2 = A^7_{12} \times B^2_{21}, \\ R^{G1}_3 &= A^8_{21} \times B^1_{12}, \; R^{G1}_4 = A^9_{21} \times B^1_{12}, \\ R^{G1}_5 &= A^{10}_{21} \times B^1_{12}, \; R^{G1}_6 = A^{11}_{21} \times B^1_{12}. \end{aligned}$$

All these attractors are confirmed by the cross graph. Computation of relative probabilities shows that the single steady state $A^{11}_{21} \times B^1_{12}$ is more frequently observed (with a percentage of around 54%, see **Table 3**). In this state all proteins of the cell cycle are expressed except for *Cdh*1 and *Sic*1, which characterize stationary G1 phase. The cell growth module is in a state where *Cln3F* is available, thus setting START to 1. The interconnected system is thus locked in a steady state where the interaction links are fixed: $A^{11}_{21} \times B^1_{12} = 11110111011 \times 10100110$, since the output of each attractor is equal to the input of the other.

TABLE 3 | Attractors of the yeast interconnected system and their relative probabilities, $P_{rel}(R_i)$, for different updating probabilities ϱ_A.

Case S-phase output			
Attractor	$\varrho_A = 0.2$	$\varrho_A = 0.5$	$\varrho_A = 0.7$
$A^8_{21} \times B^1_{12}$	0.1125	0.0938	0.0813
$A^{10}_{21} \times B^1_{12}$	0.1125	0.0938	0.0813
R^S_1	0.7750	0.8125	0.8375
Case G1-phase output			
Attractor	$\varrho_A = 0.2$	$\varrho_A = 0.5$	$\varrho_A = 0.7$
$A^4_{12} \times B^2_{21}$	0.1042	0.1266	0.1415
$A^7_{12} \times B^2_{21}$	0.0454	0.0401	0.0365
$A^8_{21} \times B^1_{12}$	0.0829	0.0691	0.0599
$A^9_{21} \times B^1_{12}$	0.1779	0.1585	0.1456
$A^{10}_{21} \times B^1_{12}$	0.0674	0.0562	0.0487
$A^{11}_{21} \times B^1_{12}$	0.5221	0.5495	0.5678

5.2.3. Hypotheses Discrimination

These results appear to support a model for START signal of the form (12), as suggested by Aldea et al. (2017). Indeed, if cell size triggers START, then it can be assumed that there is a "critical size" which will be attained most probably at the end of G2 phase. And, in fact, the interconnected system exhibits an oscillatory cycle only in the case of M phase used as cell size indicator. This cycle is in agreement with cell cycle progression, meaning that the cell size module is able to trigger the START signal.

In contrast, when G1 or S phases are used as cell size indicator, the interconnected system has no oscillatory behavior. For the G1 case, the most frequent steady state ($A^{11}_{21} \times B^1_{12}$) represents a configuration where the cell size module permanently sets $Cln3F = 1$, and does not admit cell size to reset to zero. Note that G1 is the beginning of the cell cycle and a misleading indicator of "critical" size; in this case, the "critical" size is so small that the cell size module sets START permanently to 1 thus preventing the cell cycle to reset to zero and initiate a new cycle. Cells are locked in a steady state near mitosis and before early G1.

In conclusion, our analysis shows that neither *G*1 nor S phases are reliable cell growth indicators, but components from M phase are plausible candidates for detecting cell growth. We point out that the cell size Boolean network and the feedback interconnection points may admit several improvements, which are outside the scope of our paper. Nevertheless, we believe this first approach provides useful hints on how to further investigate and model the START signal in yeast.

6. DISCUSSION AND CONCLUSIONS

Our work illustrates a new concept for the analysis of an interconnection of Boolean networks: the goal is to study the coupled behavior of two or more modules, using only the dynamics of each separate module. A new methodology has been discussed, based on construction of the asymptotic and cross graphs both representative of the full network transition graph and guaranteed to compute all attractors of the interconnected network. The two graphs have different properties but also complement each other. The cross graph provides exact results, in the sense that it contains all transient and asymptotic behaviors of the interconnected network. The asymptotic graph is a lighter construction containing a minimal number of nodes while recovering all attractors. In contrast to the cross graph, no bijection with the full network transition graph is guaranteed, implying that spurious attractors may appear; however, this happens at an extremely low rate (less than 1%).

Construction of the two graphs for random input/output networks with varying connectivity reveals their complementarity in terms of modules' connectivity: for low connectivity ($K_{mean} \leq 5$), the asymptotic graph is much smaller (on average 0.01% of the full graph, against 28% for the cross graph; **Figure 2B**) and faster to compute; in contrast, for high connectivity ($K_{mean} > 5$), the size of the cross graph drastically reduces to 0.04% of the full graph (**Figure 2A**) becoming even faster to analyze than the asymptotic graph (**Figure 2C**). In addition, even though the asymptotic graph involves a drastic simplification of the state space, it has an unexpectedly high rate of accuracy, as shown in **Figure 2D**.

The practical advantages of our methodology are illustrated by the study of two well known biological networks. Among other useful characteristics, the asymptotic graph can greatly reduce the size of the state space, especially in the case of single-input single-output modules. For instance the mammalian and yeast interconnected networks, with an average connectivity of $K = 2.76$ and $K = 2.68$ respectively, have asymptotic graphs of only 18 and 22 nodes (compared to 2^{17} or 2^{19}).

The analysis of the coupling between cell cycle and circadian clock shows that, according to experimental observations (for instance by Plikus et al., 2013), the asymptotic graph predicts that mammalian cells in the quiescent state may have a working clock. Furthermore, under general hypotheses, the probabilistic approach predicts that one third of cells are arrested in the quiescent state but still have circadian oscillations, while the other two thirds follow a normal cell cycle progression intertwined with circadian oscillations. In the budding yeast example, we have explored a recent hypothesis by Aldea et al. (2017) for a mechanism to trigger the START signal and initiate cell cycle. The mechanism is based on cell size detection through cell growth rate. Our analysis supports such a mechanism as a possible START trigger, and suggests that cell size indicator should come from an element during M phase.

The advantages of our analysis tools are multiple and particularly suited to the modeling of biological regulatory networks: by manipulating existing models as building blocks, the presented tools allow to rapidly simulate, compare, and test different coupling schemes or hypotheses on mutual regulatory effects, and therefore advance in the understanding of highly modular regulatory networks. The probabilistic interpretation and the analysis of transient behaviors emerge as two noteworthy directions for future developments in logical models.

AUTHOR CONTRIBUTIONS

MC and LT: equally contributed to conception, analysis and design of the study; MC and LT: wrote and revised the manuscript.

FUNDING

MC and LT are partly supported by the French agency for research through project ICycle ANR-16-CE33-0016-01. MC is partly funded by Labex Signalife ANR-11-LABX-0028-01.

SUPPLEMENTARY MATERIAL

The Boolean models used for mammalian and budding yeast cell cycles are provided as Supplementary Material.

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphys.2018.00586/full#supplementary-material

REFERENCES

Abou-Jaoudé, W., Traynard, P., Monteiro, P. T., Saez-Rodriguez, J., Helikar, T., Thieffry, D., et al. (2016). Logical modeling and dynamical analysis of cellular networks. *Front. Genet.* 7:94. doi: 10.3389/fgene.2016.00094

Albert, R., and Othmer, H. G. (2003). The topology of the regulatory interactions predicts the expression pattern of the *Drosophila* segment polarity genes. *J. Theor. Biol.* 223, 1–18. doi: 10.1016/S0022-5193(03)00035-3

Aldea, M., Jenkins, K., and Csikász-Nagy, A. (2017). Growth rate as a direct regulator of the start network to set cell size. *Front. Cell Dev. Biol.* 5:57. doi: 10.3389/fcell.2017.00057

Baldazzi, V., Ropers, D., Markowicz, Y., Kahn, D., Geiselmann, J., and de Jong, H. (2010). The carbon assimilation network in *Escherichia coli* is densely connected and largely sign-determined by directions of metabolic fluxes. *PLoS Comput. Biol.* 6:e1000812. doi: 10.1371/journal.pcbi.1000812

Bérenguier, D., Chaouiya, C., Monteiro, P. T., Naldi, A., Remy, E., Thieffry, D., et al. (2013). Dynamical modeling and analysis of large cellular regulatory networks. *Chaos* 23:025114. doi: 10.1063/1.4809783

Bieler, J., Cannavo, R., Gustafson, K., Gobet, C., Gatfield, D., and Naef, F. (2014). Robust synchronization of coupled circadian and cell cycle oscillators in single mammalian cells. *Mol. Syst. Biol.* 10:739. doi: 10.15252/msb.20145218

Birkhoff, G. (1940). *Lattice Theory, Vol. 25.* Cambridge, MA: American Mathematical Society.

Calzone, L., Tournier, L., Fourquet, S., Thieffry, D., Zhivotovsky, B., Barillot, E., et al. (2010). Mathematical modelling of cell-fate decision in response to death receptor engagement. *PLoS Comput. Biol.* 6:e1000702. doi: 10.1371/journal.pcbi.1000702

Chaves, M., and Carta, A. (2015). Attractor computation using interconnected boolean networks: testing growth rate models in *E. coli*. *Theor. Comput. Sci.* 599, 47–63. doi: 10.1016/j.tcs.2014.06.021

Chaves, M., and Tournier, L. (2011). "Predicting the asymptotic dynamics of large biological networks by interconnections of Boolean modules," in *Proceedings of the* 50^{th} *conference Decision and Control and European Control conference* (Orlando, FL), 3026–3031.

Comet, J.-P., Bernot, G., Das, A., Diener, F., Massot, C., and Cessieux, A. (2012). Simplified models for the mammalian circadian clock. *Proc. Comput. Sci.* 11, 127–138. doi: 10.1016/j.procs.2012.09.014

Cormen, T., Leiserson, C., Rivest, R., and Stein, C. (2001). *Introduction to Algorithms*. Providence, RI: MIT Press, McGraw-Hill.

Fauré, A., Naldi, A., Chaouiya, C., and Thieffry, D. (2006). Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle. *Bioinformatics* 22, 124–131. doi: 10.1093/bioinformatics/btl210

Feillet, C., Krusche, P., Tamanini, F., Janssens, R. C., Downey, M. J., Martin, P., et al. (2014). Phase locking and multiple oscillating attractors for the coupled mammalian clock and cell cycle. *Proc. Natl. Acad. Sci. U.S.A.* 111, 9828–9833. doi: 10.1073/pnas.1320474111

Feillet, C., van der Horst, G. T., Lévi, F., Rand, D. A., and Delaunay, F. (2015). Coupling between the circadian clock and cell cycle oscillators: implication for healthy cells and malignant growth. *Front. Neurol.* 6:96. doi: 10.3389/fneur.2015.00096

Feller, W. (1970). *An Introduction to Probability Theory and Its Applications*. New York, NY: Wiley.

García-Gómez, M., Azpeitia, E., and Álvarez-Buylla, E. (2017). A dynamic genetic-hormonal regulatory network model explains multiple cellular behaviors of the root apical meristem of *Arabidopsis thaliana*. *PLoS Comput. Biol.* 13:e1005488. doi: 10.1371/journal.pcbi.1005488

Gérard, C., and Goldbeter, A. (2012). Entrainmemt of the mammalian cell cycle by the circadian clock: modeling two coupled cellular rhythms. *PLoS Comput. Biol.* 8:e1002516. doi: 10.1371/journal.pcbi.1002516

Kauffman, S. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.* 22, 437–467. doi: 10.1016/0022-5193(69)90015-0

Klarner, H., Bockmayr, A., and Siebert, H. (2015). Computing maximal and minimal trap spaces of Boolean networks. *Nat. Comput.* 14, 535–544. doi: 10.1007/s11047-015-9520-7

Li, F., Long, T., Lu, Y., Ouyang, Q., and Tang, C. (2004). The yeast cell cycle is robustly designed. *Proc. Natl. Acad. Sci. U.S.A.* 101, 4781–4786. doi: 10.1073/pnas.0305937101

Mendoza, L., and Xenarios, I. (2006). A method for the generation of standardized qualitative dynamical systems of regulatory networks. *Theor. Biol. Med. Model.* 3, 1–18. doi: 10.1186/1742-4682-3-13

Mori, T., Flöttmann, M., Krantz, M., Akutsu, T., and Klipp, E. (2015). Stochastic simulation of boolean *rxncon* models: towards quantitative analysis of large signaling networks. *BMC Syst. Biol.* 9:45. doi: 10.1186/s12918-015-0193-8

Plikus, M. V., Vollmers, C., de la Cruz, D., Chaix, A., Ramos, R., Panda, S., et al. (2013). Local circadian clock gates cell cycle progression of transient amplifying cells during regenerative hair cycling. *Proc. Natl. Acad. Sci. U.S.A.* 110, E2106–E2115. doi: 10.1073/pnas.1215935110

Sabidussi, G. (1959). Graph multiplication. *Mathematische Zeitschrift* 72, 446–457. doi: 10.1007/BF01162967

Saez-Rodriguez, J., Simeoni, L., Lindquist, J. A., Hemenway, R., Bommhardt, U., Arndt, B., et al. (2007). A logical model provides insights into T cell receptor signaling. *PLoS Comput. Biol.* 3:e163. doi: 10.1371/journal.pcbi.0030163

Shmulevich, I., Dougherty, E. R., Kim, S., and Zhang, W. (2002). Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics* 18, 261–274. doi: 10.1093/bioinformatics/18.2.261

Stoll, G., Caron, B., Viara, E., Dugourd, A., Zinovyev, A., Naldi, A., et al. (2017). Maboss 2.0: an environment for stochastic boolean modeling. *Bioinformatics* 33, 2226–2228. doi: 10.1093/bioinformatics/btx123

Tournier, L., and Chaves, M. (2013). Interconnection of asynchronous Boolean networks, asymptotic and transient dynamics. *Automatica* 49, 884–893. doi: 10.1016/j.automatica.2013.01.015

Turner, J., Ewald, J. C., and Skotheim, J. M. (2012). Cell size control in yeast. *Curr. Biol.* 22, R350–R359. doi: 10.1016/j.cub.2012.02.041

Veliz-Cuba, A., Aguilar, B., Hinkelmann, F., and Laubenbacher, R. (2014). Steady state analysis of boolean molecular network models via model reduction and computational algebra. *BMC Bioinformatics* 15:221. doi: 10.1186/1471-2105-15-221

Wang, R. S., Saadatpour, A., and Albert, R. (2012). Boolean modeling in systems biology: an overview of methodology and applications. *Phys. Biol.* 9:055001. doi: 10.1088/1478-3975/9/5/055001

Zañudo, J., and Albert, R. (2013). An effective network reduction approach to find the dynamical repertoire of discrete dynamic networks. *Chaos* 23:025111. doi: 10.1063/1.4809777

Zhang, R., Shah, M. V., Yang, J., Nyland, S. B., Liu, X., Yun, J. K., et al. (2008). Network model of survival signaling in large granular lymphocyte leukemia. *Proc. Natl. Acad. Sci. U.S.A.* 105, 16308–16313. doi: 10.1073/pnas.0806447105

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

APPENDIX

Proof of Theorem 2

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a digraph and let $v, v' \in \mathcal{V}$ be any two vertices of $\mathcal{G}$. Introduce the following notation:

- $v \rightarrow_{\mathcal{G}} v'$ means that there is an edge from v to v' in $\mathcal{G}$, i.e., $(v, v') \in \mathcal{E}$ (v' is a *successor* of v).
- $v \rhd_{\mathcal{G}} v'$ means that there exists a path from v to v' in $\mathcal{G}$, i.e., there exist $k \geq 0$ vertices $v_1, \ldots, v_k$ such that $v = v_1 \rightarrow_{\mathcal{G}} v_2 \rightarrow_{\mathcal{G}} \ldots \rightarrow_G v_k = v'$ (v' is a *descendant* of v).
- $v \sim_{\mathcal{G}} v'$ means that there exists a path from v to v' and a path from v' to v in $\mathcal{G}$ (v and v' are mutually reachable from each other). The relation $\sim_{\mathcal{G}}$ is an equivalence over $\mathcal{V} \times \mathcal{V}$.

Remark that according to the definition of partition Z_h^A, any $\mathbf{A}_\alpha^i$ is included in a SCC of each graph $G^{A,u}$, in other words:

$$\forall a, a' \in \mathbf{A}_\alpha^i, \forall u \in \{0,1\}^{p_A}, \; a \sim_{G^{A,u}} a'.$$

For convenience, we introduce the two following maps π and ψ, establishing relationships between the two vertex sets V^{cr} and $\Omega = \{0,1\}^{n_A + n_B}$.

- For $V = \mathbf{A}_\alpha^i \times \mathbf{B}_\beta^j \in V^{cr}$, let $\pi(V) := \{(a,b) | \, a \in \mathbf{A}_\alpha^i, b \in \mathbf{B}_\beta^j\} \subseteq \Omega$; and for $Q = \{V_1, \ldots, V_k\} \subseteq V^{cr}$, define $\pi(Q) := \bigcup_{l=1}^k \pi(V_l) \subseteq \Omega$.
- For $x = (a,b) \in \Omega$, by definition of Z_h^A, Z_h^B there is a unique $\mathbf{A}_\alpha^i$ and a unique $\mathbf{B}_\beta^j$ such that $\mathbf{A}_\alpha^i \ni a$, $\mathbf{B}_\beta^j \ni b$. Let $\psi(x) := \mathbf{A}_\alpha^i \times \mathbf{B}_\beta^j$; by extension, for $S \subseteq \Omega$, define $\psi(S) := \{\psi(x) | \, x \in S\} \subseteq V^{cr}$.

Theorem 2 is a consequence of the two following lemmas.

LEMMA 1. *Let $x, y \in \Omega$ such that $x \rhd_G y$, then either $\psi(x) = \psi(y)$, or $\psi(x) \rhd_{G^{cr}} \psi(y)$.*

PROOF: Suppose first that $x \rightarrow_G y$, that is to say either (i): $x = (a,b) \rightarrow_G (a',b) = y$ where $a \rightarrow_{G^{A,h^B(b)}} a'$ or (ii): $x = (a,b) \rightarrow_G (a,b') = y$ where $b \rightarrow_{G^{B,h^A(a)}} b'$. These two cases being perfectly symmetrical, consider for instance case (i). Let $\mathbf{A}_\alpha^i$, $\mathbf{A}_{\alpha'}^{i'}$ and $\mathbf{B}_\beta^j$ be respectively the (unique) sets such that $a \in \mathbf{A}_\alpha^i$, $a' \in \mathbf{A}_{\alpha'}^{i'}$ and $b \in \mathbf{B}_\beta^j$. Two cases are to be considered.

Case 1: suppose $\mathbf{A}_\alpha^i = \mathbf{A}_{\alpha'}^{i'}$, then $\psi(x) = \mathbf{A}_\alpha^i \times \mathbf{B}_\beta^j = \psi(y)$.

Case 2: suppose $\mathbf{A}_\alpha^i \neq \mathbf{A}_{\alpha'}^{i'}$. Then according to the definition of G^{cr}, from $a \rightarrow_{G^{A,\beta}} a'$ we deduce that $\psi(x) = \mathbf{A}_\alpha^i \times \mathbf{B}_\beta^j \rightarrow_{G^{cr}} \mathbf{A}_{\alpha'}^{i'} \times \mathbf{B}_\beta^j = \psi(y)$.

Suppose now that $x \rhd_G y$, *ie.*, $x = x_1 \rightarrow_G x_2 \rightarrow_G \ldots \rightarrow_G x_k = y$. By applying successively the previous result along that path, we deduce that either $\psi(x) = \psi(y)$ or $\psi(x) \rhd_{G^{cr}} \psi(y)$, which concludes the proof. □

LEMMA 2. *Let $V, V' \in V^{cr}$ be two vertices of the cross graph.*

(i) $\forall x, y \in \pi(V)$, $x \sim_G y$.
(ii) If $V \rhd_{G^{cr}} V'$, then for all $x \in \pi(V)$ and $y \in \pi(V')$, $x \rhd_G y$.

PROOF: Let start with assertion *(i)*. Let $V = \mathbf{A}_\alpha^i \times \mathbf{B}_\beta^j$, $x = (a,b) \in \pi(V)$ and $y = (a',b') \in \pi(V)$. Since a and a' both belong to the same $\mathbf{A}_\alpha^i$, $a \sim_{G^{A,\beta}} a'$. In the same way, $b \sim_{G^{B,\alpha}} b'$. From there it is easy to verify that $(a,b) \sim_G (a',b) \sim_G (a',b')$, so $x \sim_G y$. Let us prove the second assertion. Suppose first that $V \rightarrow_{G^{cr}} V'$. For instance, let $V = \mathbf{A}_\alpha^i \times \mathbf{B}_\beta^j$ and $V' = \mathbf{A}_{\alpha'}^{i'} \times \mathbf{B}_\beta^j$ with $\mathbf{A}_\alpha^i \ni a_1 \rightarrow_{G^{A,\beta}} a_2 \in \mathbf{A}_{\alpha'}^{i'}$ (the symmetrical case can be treated completely analogously). Let $x = (a,b) \in \pi(V)$ and $y = (a',b') \in \pi(V')$. Since a and a_1 both belong to the same $\mathbf{A}_\alpha^i$, we have $a \sim_{G^{A,\beta}} a_1$. Similarly $a', a_2 \in \mathbf{A}_{\alpha'}^{i'}$, implying $a' \sim_{G^{A,\beta}} a_2$. Therefore we have $a \sim_{G^{A,\beta}} a_1 \rightarrow_{G^{A,\beta}} a_2 \sim_{G^{A,\beta}} a'$, hence $(a,b) \rhd_G (a',b)$. Now, since b and b' both belong to the same $\mathbf{B}_\beta^j$ $b \sim_{G^{B,\alpha'}} b'$, which proves that $(a',b) \rhd_G (a',b')$, therefore $x \rhd_G y$.

Suppose now that $V \rhd_{G^{cr}} V'$, *ie.*, $V = V_1 \rightarrow_{G^{cr}} V_2 \rightarrow_{G^{cr}} \ldots \rightarrow_{G^{cr}} V_k = V'$. By applying successively the previous result along that path, we deduce that $x \rhd_G y$ for any $x \in \pi(V)$ and $y \in \pi(V')$. □

Lemmas 1 and 2 establish an exact correspondence between the paths in G and the paths in G^{cr}. The proof of the theorem becomes rather straightforward. Indeed, suppose $Q = \{V_1, \ldots, V_k\}$ is a SCC of G^{cr}. Then Lemma 2 implies that $\pi(Q)$ is included in a SCC S of G. Suppose now that $\pi(Q) \subsetneq S$, *ie.* there exists $y \in S \backslash \pi(Q)$ such that $\psi(y) \notin Q$. For any $x \in \pi(V) \subset S$, we have $x \sim_G y$ (with Lemma 1), implying $\psi(y) \in Q$ which is a contradiction. Therefore, $\pi(Q) = S$. Reciprocally, suppose $S \subseteq \Omega$ is a SCC of G. Then lemma 1 implies that $\psi(S)$ is included in a SCC Q of G^{cr}. Lemma 2 further yields $\psi(S) = Q$. By using a similar kind of reasoning, it is easy to show that there is an exact one-to-one correspondence between the terminal SCCs of G^{cr} and the attractors of G.

frontiers in Physiology

METHODS
published: 08 June 2018
doi: 10.3389/fphys.2018.00695

Identification of Boolean Network Models From Time Series Data Incorporating Prior Knowledge

*Thomas Leifeld, Zhihua Zhang and Ping Zhang**

Institute of Automatic Control, Technische Universität Kaiserslautern, Kaiserslautern, Germany

Motivation: Mathematical models take an important place in science and engineering. A model can help scientists to explain dynamic behavior of a system and to understand the functionality of system components. Since length of a time series and number of replicates is limited by the cost of experiments, Boolean networks as a structurally simple and parameter-free logical model for gene regulatory networks have attracted interests of many scientists. In order to fit into the biological contexts and to lower the data requirements, biological prior knowledge is taken into consideration during the inference procedure. In the literature, the existing identification approaches can only deal with a subset of possible types of prior knowledge.

Results: We propose a new approach to identify Boolean networks from time series data incorporating prior knowledge, such as partial network structure, canalizing property, positive and negative unateness. Using vector form of Boolean variables and applying a generalized matrix multiplication called the semi-tensor product (STP), each Boolean function can be equivalently converted into a matrix expression. Based on this, the identification problem is reformulated as an integer linear programming problem to reveal the system matrix of Boolean model in a computationally efficient way, whose dynamics are consistent with the important dynamics captured in the data. By using prior knowledge the number of candidate functions can be reduced during the inference. Hence, identification incorporating prior knowledge is especially suitable for the case of small size time series data and data without sufficient stimuli. The proposed approach is illustrated with the help of a biological model of the network of oxidative stress response.

Conclusions: The combination of efficient reformulation of the identification problem with the possibility to incorporate various types of prior knowledge enables the application of computational model inference to systems with limited amount of time series data. The general applicability of this methodological approach makes it suitable for a variety of biological systems and of general interest for biological and medical research.

Keywords: Boolean networks, identification, prior knowledge, time series data, network inference

OPEN ACCESS

Edited by:
Tomáš Helikar,
University of Nebraska-Lincoln,
United States

Reviewed by:
Aurélien Naldi,
École Normale Supérieure, France
Ruisheng Wang,
Department of Medicine, Brigham and Women's Hospital, United States

****Correspondence:***
Ping Zhang
pzhang@eit.uni-kl.de

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: *01 February 2018*
Accepted: *18 May 2018*
Published: *08 June 2018*

Citation:
Leifeld T, Zhang Z and Zhang P (2018) Identification of Boolean Network Models From Time Series Data Incorporating Prior Knowledge. Front. Physiol. 9:695. doi: 10.3389/fphys.2018.00695

1. INTRODUCTION

Boolean networks (BNs) are discrete-time systems, whose variables can take only two possible values (i.e., 0 and 1). Since Stuart Kaufman firstly introduced BNs in Kauffman (1969) for qualitative description of gene regulatory interactions, BNs have attracted great attention from many scientists and several results have been proposed, for instance, analysis (Albert and Barabási, 2000) and control (Fauré et al., 2006). An overview can be found in Wang et al. (2012) and a database for established models and compatible tools has been introduced (Naldi et al., 2015).

Mathematical models are important to explain dynamic behavior of a system and to understand the functionality of system components (Grieb et al., 2015) and can help scientists to design model-based targeted therapy and diagnosis (Fumia and Martins, 2013). Hence, the inference of models capturing the relevant behavior of the system is an important topic. The inference can be based on the connection of known biochemical reactions, like BN model for the yeast cell cycle in Davidich and Bornholdt (2008), or on experimental data, if the latter is the case it is also called the identification problem. One of the first approaches to identify a BN was REVEAL which is based on mutual information (Liang et al., 1998). In Akutsu et al. (1999) a similar but less complex approach is presented. Both cannot handle errors in the dataset which was solved in Lähdesmäki et al. (2003). The modeled quantities are not Boolean in the experimental data and need to be binarized first. For the binarization several approaches can be found in the literature ranging from mixture model based clustering (Zhou et al., 2003) to more complex methods where the significance of a jump in the time series is estimated in Hopfensitz et al. (2012). A comparison of some identification and binarization approaches and their combinations can be found in Berestovsky and Nakhleh (2013). Most identification approaches are based on previously binarized data, but there also exist approaches directly based on continuous data (e.g., Karlebach and Shamir, 2012). In Higa et al. (2011) the data is considered as given constraint and the set of systems fulfilling the constraints is searched. This approach was then further improved by reducing the sensitivity to noise in Ouyang et al. (2014). An example of recent research is the identification of Boolean models for transient dynamics after perturbations from time course data with answer set programming (Ostrowski et al., 2016). A BN can simply be extended to a Boolean control network (BCN) by considering manipulated external stimuli as control signal of the network. Recently, a powerful tool called semi-tensor product (STP) of matrices has been proposed in Cheng (2001), which can convert the dynamics of BCNs into a model where all information of the dynamics and the structure of the BCN is contained in two matrices (Cheng et al., 2011a). Using the STP based matrix description of BCN several approaches for identifying BCN have been proposed (Cheng and Zhao, 2011; Fornasini and Valcher, 2014; Zhang et al., 2017a).

However, in general, in order to identify the dynamical model of a BCN from its input and output data, a huge number of data is required (Cheng and Zhao, 2011; Cheng et al., 2011b). Though, in practice, data size is limited by the cost of experiments (Geier et al., 2007). In order to reduce the search space and improve the accuracy of the model, the benefit of biological prior knowledge should be taken into consideration. Cheng and Zhao (2011) pointed out that, if the network graph is known, then the data required can be reduced considerably. In the literature there are several approaches to include different types of prior knowledge. For example the known network structure and known steady state activity is considered in Videla et al. (2015). Moreover, two common properties of the Boolean function, canalizing and unateness, can be further utilized according to Breindl et al. (2013) and Faisal et al. (2010). A Boolean function is canalizing, if a variable takes on a certain "canalizing" value, then the output of the boolean function is always the same (Waddington, 1942). Different from canalizing function, an unate function has monotonic properties, which in biology indicates that a gene acts exclusively as an inducer or as an inhibitor for the expression of another gene (Porreca et al., 2010). The prior knowledge is used in different ways either by introducing additional constraints in the optimization (Breindl et al., 2013), or reducing the number of parameters in the optimization (Cheng and Zhao, 2011). In Dorier et al. (2016) and Terfve et al. (2012) genetic algorithms are used to handle the complexity problem of large networks while satisfying prior knowledge network graphs as constraints. However, these approaches to handle prior knowledge are not compatible and the advantages of different types of prior knowledge can not be combined. In the approach proposed in this paper, all different types of prior knowledge can be utilized simultaneously and it can additionally handle hypotheses for interactions, which could be used for researcher bias free distinction between alternative hypotheses. Furthermore existing approaches can not handle the case that at some time instances some measurement values are missing, which cannot be avoided in practice due to the limitation of measuring techniques like mass spectrometry-based proteomics.

In this paper, we consider the identification problem of BCNs utilizing biological prior knowledge. A part of the results was presented at the 56th IEEE Conference on Decision and Control in Melbourne (Zhang et al., 2017b). However, the BCN model considered in Zhang et al. (2017b) contains a general output equation. By applying prediction error method (PEM), a high-dimensional BCN (i.e., $2^n \times 2^{n+m}$) cannot be avoided. Different from that, although the handling of unmeasurable processes is considered in this paper, the proposed approach leads to a low-dimensional matrix for PEM. Besides, more prior biological knowledge is considered in the paper, like potential interactions, known attractors and limit cycles. Moreover, it is discussed how to deal with alternative hypotheses for interactions and missing measurement points. The main contributions of this paper are as follows:

- A suitable way to handle the prior knowledge such as known network graph, hypotheses for interactions, canalizing and unateness properties or attractor is introduced. For this purpose the BCN is described by two matrices with unknown parameters as entries. If possible, some parameters are inferred directly. Otherwise, relationships between the parameters are set up.
- An approach to deal with the identification of BCNs, in particular, from noisy measurements and missing data points

is proposed. The identification problem of BCNs is formulated as a nonlinear pseudo-Boolean optimization, which can be equivalently transformed into a linear binary optimization problem and then solved efficiently.

The remainder of the paper is organized as follows. Section 2 introduces some fundamental definitions and notations. In Section 3, the identification problem of BCNs addressed in this paper will be formulated. Section 4 introduces a way to utilize prior knowledge in identification procedure. The formulation of identification problem of BCNs as an integer linear programming problem is derived and an example is given in Section 5 to illustrate the approach. Finally, a short discussion on the advantages and limitations of the proposed approach is given in Section 6.

2. PRELIMINARIES

In this part, we list some necessary notations, which will be used in the subsequent sections.

1. $\neg$, $\wedge$ and $\vee$ denote the logical negation (not), conjunction (and) and disjunction (or), respectively.
2. $\mathcal{D} := \{1, 0\}$ and $\mathcal{D}^n = \underbrace{\mathcal{D} \times \mathcal{D} \times \cdots \times \mathcal{D}}_{n}$.
3. $\Delta_n := \{\delta_n^k | 1 \leq k \leq n\}$, where δ_n^k denotes the k-th column of the identity matrix I_n.
4. For a vector $v \in \mathbb{R}^m$, its j-th entry is denoted by $[v]_j, j = 1, 2, \cdots, m$.
5. An $n \times t$ matrix L is called a logical matrix, if $L = [\delta_n^{i_1}\ \delta_n^{i_2}\ \cdots\ \delta_n^{i_t}]$, where $i_1, i_2, \cdots, i_t \in \{1, 2, \cdots, n\}$, and we express L briefly as $L = \delta_n[i_1\ i_2\ \cdots\ i_t]$. Denote the set of $n \times t$ logical matrices by $\mathcal{L}_{n\times t}$. $Col_i(M)$ denotes the i-th column of the matrix M.
6. $\mathbf{0}_n := [\underbrace{0\ 0\ \cdots\ 0}_{n}]^{\mathrm{T}}$, where the superscript T denotes the transpose.

The concept of the semi-tensor product of matrices (STP) has been introduced by Cheng et al. (2011a). The STP of two matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$ is defined as

$$A \ltimes B = (A \otimes I_{l/n}) \cdot (B \otimes I_{l/p}) \tag{1}$$

where $\otimes$ is the Kronecker product and $l = lcm\{n, p\}$ is the least common multiple of n and p. The following property of the STP will be used in the subsequent sections.

Lemma 1. *Let $X \in \mathbb{R}^{m\times 1}$ and $Y \in \mathbb{R}^{n\times 1}$. Then $Y \ltimes X = W_{[m,n]} \ltimes X \ltimes Y$, where $W_{[m,n]}$ is the swap matrix (Cheng et al., 2011a).*

So the order of two vectors which are multiplied can be altered by multiplying a suitable matrix from the left, this is also called the pseudo-commutativity of the STP. In the following parts the symbol $\ltimes$ will be omitted.

3. PROBLEM FORMULATION

System identification is the determination of a model describing the dynamic behavior of a system based on measured data and known perturbations. In the context of Boolean modeling it is assumed that the transient behavior of the system can be qualitatively described by a finite number of Boolean states and that the interaction of these states can be described by Boolean functions. The perturbations are inputs to the system and cause transient behavior of the interacting states in the system. A measured time series of inputs and states form together the data basis for the identification. Depending on the system which is to be modeled, the states might represent the activity of genes or the abundance of proteins and the perturbations could be a stress like heat or oxygen or a chemical substance. In the following the identification process will be formulated as mathematical optimization problem. Therefore the mathematical model of a BCN needs to be defined first. A Boolean control network (BCN) can be described by the following equations (Cheng and Qi, 2010):

$$\begin{cases} X_1(t+1) &= f_1(X_1(t), \cdots, X_n(t), U_1(t), \cdots, U_m(t)) \\ &\vdots \\ X_n(t+1) &= f_n(X_1(t), \cdots, X_n(t), U_1(t), \cdots, U_m(t)) \end{cases} \tag{2}$$

where $X(t) = [X_1(t)\ X_2(t)\ \cdots\ X_n(t)]^{\mathrm{T}} \in \mathcal{D}^n$, $U(t) = [U_1(t)\ U_2(t)\ \cdots\ U_m(t)]^{\mathrm{T}} \in \mathcal{D}^m$ are, respectively, the state vector, input vector at time t, f_i are logic functions. At the discrete time instances t the state variables are updated synchronously according to the logic functions f_i. As shown in Cheng and Qi (2010), a vector form of Boolean variable $X_i, i = 1, 2, \cdots, n$ can be simply expressed as

$$x_i = \begin{bmatrix} X_i \\ \neg X_i \end{bmatrix}. \tag{3}$$

Let $x = \ltimes_{i=1}^n x_i \in \Delta_{2^n}, u = \ltimes_{i=1}^m u_i \in \Delta_{2^m}$. According to Cheng and Qi (2010), (2) can be equivalently represented in a vector form:

$$\begin{cases} x_1(t+1) = S_1 u(t) x(t) \\ \quad\vdots \\ x_n(t+1) = S_n u(t) x(t) \end{cases}, \tag{4}$$

where $S_i \in \mathcal{L}_{2\times 2^{n+m}}, i = 1, 2, \cdots, n$ are logical matrices. Multiplying all Equations in (4) together, there is

$$x(t+1) = Lu(t)x(t) \tag{5}$$

where $L \in \mathcal{L}_{2^n\times 2^{n+m}}$ is a logical matrix and $Col_i(L) = \ltimes_{j=1}^n Col_i(S_j), i = 1, 2, \cdots, 2^{n+m}$.

A polynomial $P_{ml} : \mathbb{R}^k \to \mathbb{R}$ with k variables $\{\theta_1, \theta_2, \cdots, \theta_k\}$ is called multi-linear polynomial, if its degree in each variable is at most 1 (Alon et al., 1991). So, a multi-linear polynomial can be generally expressed as

$$P_{ml}(\theta_1, \theta_2, \cdots, \theta_k) = c + \sum_{i=1}^{k} c_i\theta_i + \sum_{\alpha=1}^{q} c_{\mathcal{I}_\alpha} \prod_{j\in\mathcal{I}_\alpha} \theta_j \tag{6}$$

where $c, c_i, c_{\mathcal{I}_\alpha} \in \mathbb{R}$ for $\mathcal{I}_\alpha \subset V = \{1, 2, \cdots, k\}$ and the set $\mathcal{I}_\alpha$ has a cardinality of at least 2, i.e., $|\mathcal{I}_\alpha| \geq 2, \alpha = 1, 2, \cdots, q$.

Generally, the identification problem of BCNs can be described as reconstruction of Boolean functions $f_i, i = 1, 2, \cdots, n$ that explain the experimental data as well as possible. Because of equivalent representation of a Boolean function by a logical matrix, the identification problem is reformulated as searching for logical matrices $S_i \in \mathcal{L}_{2\times 2^{n+m}}, i = 1, 2, \cdots, n$ based on the input and measurement state data.

Note that any logical matrix in $\mathcal{L}_{2^a\times 2^b}$ can be expressed by multi-linear polynomials in a binary parameter vector $\boldsymbol{\theta}$ of dimension $a \cdot 2^b$. For example, any logical matrix in $\mathcal{L}_{4\times 8}$ can be expressed by a binary parameter vector $\boldsymbol{\theta} = [\theta_1 \ \theta_2 \ \cdots \ \theta_{16}]^{\mathrm{T}}$ as

$$\begin{bmatrix} \theta_1\cdot\theta_2 & \theta_1\cdot(1-\theta_2) & (1-\theta_1)\cdot\theta_2 & (1-\theta_1)\cdot(1-\theta_2) \\ \theta_3\cdot\theta_4 & \theta_3\cdot(1-\theta_4) & (1-\theta_3)\cdot\theta_4 & (1-\theta_3)\cdot(1-\theta_4) \\ \theta_5\cdot\theta_6 & \theta_5\cdot(1-\theta_6) & (1-\theta_5)\cdot\theta_6 & (1-\theta_5)\cdot(1-\theta_6) \\ \theta_7\cdot\theta_8 & \theta_7\cdot(1-\theta_8) & (1-\theta_7)\cdot\theta_8 & (1-\theta_7)\cdot(1-\theta_8) \\ \theta_9\cdot\theta_{10} & \theta_9\cdot(1-\theta_{10}) & (1-\theta_9)\cdot\theta_{10} & (1-\theta_9)\cdot(1-\theta_{10}) \\ \theta_{11}\cdot\theta_{12} & \theta_{11}\cdot(1-\theta_{12}) & (1-\theta_{11})\cdot\theta_{12} & (1-\theta_{11})\cdot(1-\theta_{12}) \\ \theta_{13}\cdot\theta_{14} & \theta_{13}\cdot(1-\theta_{14}) & (1-\theta_{13})\cdot\theta_{14} & (1-\theta_{13})\cdot(1-\theta_{14}) \\ \theta_{15}\cdot\theta_{16} & \theta_{15}\cdot(1-\theta_{16}) & (1-\theta_{15})\cdot\theta_{16} & (1-\theta_{15})\cdot(1-\theta_{16}) \end{bmatrix}^{\mathrm{T}}$$

where the superscript T denotes the transpose. In this way, each realization of the binary parameter vector $\boldsymbol{\theta} \in \mathcal{D}^{a2^b}$ corresponds to a unique logical matrix. It is straightforward to equivalently convert this logical matrix into readable logical equations. Based on this, the objective of the paper is to find a binary parameter vector $\boldsymbol{\theta}$, such that dynamic behavior of the BCN (5) is consistent with the important dynamics captured in the observed input-state data.

4. INCORPORATION OF PRIOR KNOWLEDGE

In this section, we shall show how to utilize known network graph, potential interactions, canalizing and unateness properties and attractors in the identification procedure.

4.1. Known or Potential Interactions

Often some or all interaction partners are known in a biological system which is subject of identification. This knowledge can come from databases or can be constructed based knowledge about the underlying biochemical reactions. In some cases a known signaling network is to be complemented and different hypothesis for potential interactions shall be evaluated. If all interaction partners and the direction of the interactions are known, the underlying directed network graph of the BN is known.

In graph theory, a directed graph can be denoted by $G = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is a finite set of nodes and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is a finite set of edges (Bollobas, 2012). If $(v_i, v_j) \in \mathcal{E}$, then there is an edge from $v_i \to v_j$. According to Cheng et al. (2011a), a BCN can be represented by a directed graph, where each gene is considered as a node. If there is an edge from $X_i \to X_j$, then X_j is affected by X_i.

Assume that a directed graph for a BCN $G = \{\mathcal{V}, \mathcal{E}\}$ is known. Then we have the following result.

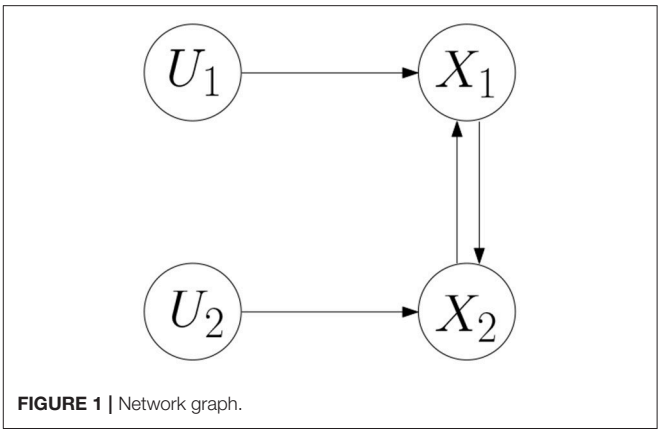


FIGURE 1 | Network graph.

Lemma 2. *If the node X_i is affected by w nodes, then 2^w binary parameters are enough to describe the corresponding logical matrix S_i.*

Proof: As the node x_i is affected by w nodes, then the Boolean function can be represented in a vector form as

$$x_i(t+1) = S_i x_{i_1}(t) x_{i_2}(t) \cdots x_{i_w}(t)$$

where the matrix S_i is a logical matrix of dimension 2×2^w. Recall that the logical matrix S_i is a matrix containing only columns belonging to Δ_2 (Cheng et al., 2011a). Hence, 2^w binary parameters are enough for the description of the logical matrix S_i.

An example is given below to express logical matrices of a BCN with a known network graph with the help of binary parameters.

Example 1. Consider a BCN as follows.

$$\begin{cases} X_1(t+1) &= f_1(X_2(t), U_1(t)) \\ X_2(t+1) &= f_2(X_1(t), U_2(t)) \end{cases} \tag{7}$$

where the network graph of the BCN is shown in **Figure 1** (Cheng and Zhao, 2011). According to Cheng and Qi (2010), the algebraic form of the BCN is obtained,

$$\begin{cases} x_1(t+1) = S_1 u_1(t) x_2(t) \\ x_2(t+1) = S_2 u_2(t) x_1(t) \end{cases} \tag{8}$$

where the logical matrices $S_1, S_2 \in \mathcal{L}_{2\times 4}$ can be expressed by the binary parameter vector $\boldsymbol{\theta} = [\theta_1 \ \theta_2 \ \cdots \ \theta_8]^{\mathrm{T}}$ in the following form:

$$S_1 = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 \\ 1-\theta_1 & 1-\theta_2 & 1-\theta_3 & 1-\theta_4 \end{bmatrix},$$
$$S_2 = \begin{bmatrix} \theta_5 & \theta_6 & \theta_7 & \theta_8 \\ 1-\theta_5 & 1-\theta_6 & 1-\theta_7 & 1-\theta_8 \end{bmatrix}.$$

Potential interactions can be treated in the same way as known interactions as long as all of them could potentially be simultaneously true. If there are two alternative hypotheses and the question is which fits better to the data, then this can be done by introducing a constraint on the parameters θ.

Example 2. Assume that X_1 is influenced either by X_2 or by U_1, this could be ensured by imposing the constraint

$$\lambda(\theta_1-\theta_2)\cdot(\theta_3-\theta_4)+(1-\lambda)(\theta_1-\theta_3)\cdot(\theta_2-\theta_4)=0,\ \ \lambda\in\{0,1\}, \tag{9}$$

4.2. Canalizing Boolean Functions

The concept of "canalizing" values in Boolean functions was introduced in developmental biology in 1940s (Waddington, 1942). The idea is, that one input is dominant and if it takes a certain value it determines the output. After that, in order to explain the phenomenon that absence of repressor or high levels of allolactose assures the operator cannot bind repressor in *lac operon* of the bacterium *Escherichia coli*, Kauffman applied this concept to BN modeling of gene regulatory networks (Kauffman, 1974).

Canalizing functions are defined as follows.

Definition 1. A Boolean function $f:\mathcal{D}^n \xrightarrow{f} \mathcal{D}$ is canalizing if there exist a variable $X_i, i\in\{1,2,\cdots,n\}$ and a Boolean function $g(X_1,\cdots,X_{i-1},X_{i+1},\cdots,X_n)$ and $a,b\in\mathcal{D}$, such that

$$f(X_1,\cdots,X_n)=\begin{cases} b, & \text{if } X_i=a,\\ g\neq b, & \text{if } X_i\neq a\end{cases} \tag{10}$$

where a is called the canalizing value for the variable X_i and b is the canalizing output value (Kauffman, 1974).

Based on Definition 1, this prior knowledge can be translated into imposing a specified value in the corresponding logical matrix. Assume that the logical matrix for the canalizing function (10) is denoted as S and the canalizing value a and canalizing output b can, respectively, be expressed in a vector form as δ_2^{2-a} and δ_2^{2-b}. Then, we can get the following result.

Theorem 1. *Given a canalizing function (10). The corresponding logical matrix $S\in\mathcal{L}_{2\times 2^n}$ satisfies*

$$SW_{[2,2^{i-1}]}\delta_2^{2-a}=\delta_2[\underbrace{2-b\ \ 2-b\ \cdots\ 2-b}_{2^{n-1}}]. \tag{11}$$

where $W_{[2,2^{i-1}]}$ is the swap matrix.

Proof: According to Lemma 1, it is easy to obtain $Sx_1x_2\cdots x_n = SW_{[2,2^{i-1}]}x_ix_1x_2\cdots x_{i-1}x_{i+1}\cdots x_n$. Applying (11), we have

$$\begin{aligned}&SW_{[2,2^{i-1}]}\delta_2^{2-a}x_1x_2\cdots x_{i-1}x_{i+1}\cdots x_n\\ &=\delta_2[\underbrace{2-b\ \ 2-b\ \cdots\ 2-b}_{2^{n-1}}]x_1x_2\cdots x_{i-1}x_{i+1}\cdots x_n=\delta_2^{2-b}\end{aligned}$$

which corresponds to $f(X_1,\cdots,X_{i-1},a,X_{i+1},\cdots,X_n)=b$ for any $X_1,\cdots,X_{i-1},X_{i+1},\cdots,X_n\in\{0,1\}$.

Let's take an example to illustrate the result of Theorem 1.

Example 3. Consider the BCN (7). Assume that the Boolean function f_1 is a canalizing function in x_2 for a canalizing value δ_2^2 and the corresponding canalizing output is δ_2^1. Due to the canalizing property, the logical matrix S_1 can be reduced to

$$S_1W_{[2,2]}\delta_2^2=\begin{bmatrix}1&1\\0&0\end{bmatrix}\Rightarrow S_1=\begin{bmatrix}\theta_1&1&\theta_3&1\\1-\theta_1&0&1-\theta_3&0\end{bmatrix}.$$

It can be checked that $S_1u_1\delta_2^2=\delta_2^1$, no matter whether $u_1=\delta_2^1$ or $u_1=\delta_2^2$. Note that the logical matrix S_1 contains only two binary parameters (i.e., θ_1 and θ_3). It shows that using canalizing property can reduce the number of binary parameters.

As an important subclass of canalizing function, k-canalizing function is defined as follows.

Definition 2. Let σ be a permutation on the set $\{1,2,\cdots,n\}$. A Boolean function $f:\mathcal{D}^n \xrightarrow{f} \mathcal{D}$ is k-canalizing in the variable order $X_{\sigma(1)},X_{\sigma(2)},\cdots,X_{\sigma(k)}$ with canalizing input values $a_1,a_2,\cdots,a_k$ and canalizing output values $b_1,b_2,\cdots,b_k$, if it can be represented in the form (Kauffman et al., 2003).

$$f(X_1,\cdots,X_n)=\begin{cases} b_1, & \text{if } X_{\sigma(1)}=a_1,\\ b_2, & \text{if } X_{\sigma(1)}\neq a_1, X_{\sigma(2)}=a_2,\\ \vdots & \\ b_k, & \text{if } X_{\sigma(1)}\neq a_1, X_{\sigma(2)}\neq a_2\cdots\\ & \ \ X_{\sigma(k)}=a_k,\\ g\neq b_k, & \text{if } X_{\sigma(1)}\neq a_1, X_{\sigma(2)}\neq a_2\cdots\\ & \ \ X_{\sigma(k)}\neq a_k.\end{cases} \tag{12}$$

Note that if all variables have certain canalizing values, then the function is called *nested canalizing function* (Kauffman et al., 2003).

As a Boolean variable can only take two values, i.e., $\{0,1\}$, (12) can be equivalently expressed as $f(X_1,\cdots,X_n)=b_i$, if $X_{\sigma(1)}=1-a_1, X_{\sigma(2)}=1-a_2,\cdots,X_{\sigma(i)}=a_k, i=1,2,\cdots,k$. Using the Boolean variables $[X_{\sigma(1)}\,X_{\sigma(2)}\cdots X_{\sigma(i)}]^{\mathrm{T}}$ to represent a multi-valued logic variable, it is straightforward to recognize that a k-canalizing function can be equivalently formulated as a canalizing function in a multi-valued logic variable. Therefore, Theorem 1 can be applied to specify the logical matrix for k-canalizing or nested canalizing function (12).

It is necessary to point out that different from the approaches proposed in Breindl et al. (2013) and Faisal et al. (2010), some binary parameters can be directly inferred, no matter which canalizing value the canalizing variable takes.

Example 4. Consider the BCN (7). Assume that the Boolean function f_2 is nested canalizing function, which can be represented as

$$f_2(U_2,X_1)=\begin{cases}1, & \text{if } U_2=1,\\ 0, & \text{if } U_2\neq 1, X_1=1.\end{cases}$$

Because $f_2(1, X_1) = 1$ for $X_1 \in \{0, 1\}$, we have

$$S_2\delta_2^1 = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \Rightarrow S_2 = \begin{bmatrix} 1 & 1 & \theta_7 & \theta_8 \\ 0 & 0 & 1-\theta_7 & 1-\theta_8 \end{bmatrix}.$$

Moreover, due to $f_2(0, 1) = 0$, there is

$$S_2\delta_2^2\delta_2^1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow S_2 = \begin{bmatrix} 1 & 1 & 0 & \theta_8 \\ 0 & 0 & 1 & 1-\theta_8 \end{bmatrix}.$$

Remark 1. *Theorem 1 implies that considering canalizing property of a Boolean function, the corresponding logical matrix can be expressed with fewer binary parameters. For instance, if a Boolean function $f(X_1, X_2, \cdots, X_n)$ is a k-canalizing function, then 2^{n-k} different binary parameters are enough to represent the corresponding logical matrix.*

4.3. Unate Boolean Functions

The behavior of some substances or genes are well studied and it is known that they act as suppressing or activating in all reactions they are involved. If they always act inhibiting they have the so called negative unateness property. For the case that a quantity exclusively induces the expression of another gene or substance it has the positive unateness property (Porreca et al., 2010).

For the mathematical modeling of the unatess properties let us consider another important type of Boolean functions, which is called the unate function (Breindl et al., 2013).

Definition 3. (Breindl et al., 2013) A Boolean function $f : \mathcal{D}^n \xrightarrow{f} \mathcal{D}$ is unate in x_i, if for any $[X_1\ X_2 \cdots X_{i-1}\ X_{i+1} \cdots X_n]^{\mathrm{T}} \in \mathcal{D}^{n-1}$ it holds for positive unateness that

$$f(\cdots, X_{i-1}, 0, X_{i+1}, \cdots) \leq f(\cdots, X_{i-1}, 1, X_{i+1}, \cdots) \tag{13}$$

or it always holds for negative unateness that

$$f(\cdots, X_{i-1}, 0, X_{i+1}, \cdots) \geq f(\cdots, X_{i-1}, 1, X_{i+1}, \cdots) \tag{14}$$

In the same way as Breindl et al. (2013), unateness can be equivalently represented as linear formulation. Afterwards, this linear formulation can be seen as additional inequality constraints in the optimization problem. As Boolean function can be rewritten as a vector form (4) and according to Lemma 1, there is

$$Sx_1x_2 \cdots x_{i-1}x_ix_{i+1} \cdots x_n = SW_{[2,2^{i-1}]}x_ix_1x_2 \cdots x_{i-1}x_{i+1} \cdots x_n \tag{15}$$

where S is the logical matrix corresponding to the Boolean function f. Hence, $f(\cdots, X_{i-1}, 0, X_{i+1}, \cdots)$ and $f(\cdots, X_{i-1}, 1, X_{i+1}, \cdots)$ can, respectively, be represented in a vector form as

$$SW_{[2,2^{i-1}]}\delta_2^2x_1x_2 \cdots x_{i-1}x_{i+1} \cdots x_n \tag{16}$$

and

$$SW_{[2,2^{i-1}]}\delta_2^1x_1x_2 \cdots x_{i-1}x_{i+1} \cdots x_n \tag{17}$$

Furthermore, based on the vector form of Boolean variable (3) and according to (13) or (14), for each $x_1, x_2, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n \in \Delta_2$ an inequality can be set up. Putting all inequality constraints together, we can find a matrix A for the following expression.

$$A \cdot \boldsymbol{\theta} \leq \mathbf{0}_n \tag{18}$$

Example 5. Consider the Boolean function $x_1 = f_1(x_2)$, this function f_1 is defined by two unknown parameters θ_1 and θ_2 . Assume that the Boolean function f_1 is a unate function with respect to x_2, which satisfies (13). As the first step, the matrix $S_1\delta_2^1$ and $S_1\delta_2^2$ are calculated, which yields

$$S_1\delta_2^1 = \begin{bmatrix} \theta_1 \\ 1-\theta_1 \end{bmatrix},\ S_1\delta_2^2 = \begin{bmatrix} \theta_2 \\ 1-\theta_2 \end{bmatrix}.$$

Then, the inequality constraint is

$$\theta_2 \leq \theta_1 \Longleftrightarrow \begin{bmatrix} -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \leq 0.$$

4.4. Known Attractors or Limit Cycles

When the BCN is not perturbed for a sufficiently long time it reaches the steady state. The steady state of a BCN can be exactly one state (i.e., attractor) or a fix cycle of some states (i.e., limit cycle). Attractors or limit cycles are assumed to determine the phenotype in the cell differentiation (Huang and Ingber, 2000). The experimental setup to measure the steady state of a system is simpler and measurements are easier to reproduce compared with transient dynamics. As a result, the steady state of the BN is often already known when the perturbation experiments for identification of the transient behavior are carried out. This knowledge can be utilized as follows.

An attractor corresponds to a self loop in the reachability graph. For a given input combination this fixes one specific coulumn in the matrix L. For the constant input $u(t) = \delta_{2^m}^i$ and the constant state $x(t) = \delta_{2^n}^j$ the k-th column is known to be $Col_k(L) = \delta_{2^n}^j$ with $k = (i-1)2^n + j$. A limit cycle can be analyzed in a similar manner. For the given state sequence of the limit cycle of length T and the constant input $u(t) = \delta_{2^m}^i$ one can calculate T columns of L. For each time instant t of the cycle the actual state $x(t) = \delta_{2^n}^j$ and the next state $x(t+1) = \delta_{2^n}^w$ is known. The information of this known transition is used by setting the k-th column to $Col_k(L) = \delta_{2^n}^w$ with $k = (i-1)2^n + j$.

5. IDENTIFICATION APPROACH

In this part, the identification problem of BCNs will be studied. At first, it will be shown that the identification problem can be reformulated as a nonlinear pseudo-Boolean optimization problem by applying the idea of the prediction error method.

The pseudo-Boolean optimization can be transformed into an equivalent linear binary integer programming problem that can be solved more efficiently. Then, we give a way to deal with missing measurement values. Finally, we discuss how dependencies between measured substances can be handled.

5.1. Optimization Problem

The prediction error method (PEM) is one of the most widely used identification methods (Isermann and Münchhof, 2011). The basic idea behind this method is to choose parameters to make the difference between a prediction based on the model and the measured values as small as possible. As the PEM minimizes the prediction error in the identified system, errors in the data set due to noise need no special treatment. Obviously the more noise is expected in the data set the more data should be acquired for identification of a reliable model.

Before applying PEM, it is necessary to specify a measure of prediction error. In information theory, the Hamming distance $d(X, Y)$ between two vectors $X, Y \in \mathcal{D}^n$ is defined as the number of positions, in which the entries differ (Hamming, 1950).

$$d(X, Y) = |\{j \in \{1, 2, \cdots, n\} | \, [X]_j \neq [Y]_j\}| \tag{19}$$

As each entry in the vectors X and Y belongs to the Boolean domain $\{0, 1\}$, (19) can be equivalently written as

$$d(X, Y) = \sum_{i=1}^{n} \left| [X]_i - [Y]_i \right| \tag{20}$$

Furthermore, let x_i, y_i be, respectively, the vector form of $[X]_i$ and $[Y]_i$. Then, it is straightforward to get

$$\left| [X]_i - [Y]_i \right| = 1 - x_i^{\mathrm{T}} \cdot y_i \tag{21}$$

Based on this, the Hamming distance $d(X, Y)$ can be rewritten as

$$d(X, Y) = \sum_{i=1}^{n} \left(1 - x_i^{\mathrm{T}} \cdot y_i \right) \tag{22}$$

Assume that the observed input and state data is $\{(U(t), X(t)), t = 0, 1, \cdots, T\}$. The vector form of the input data $\{U_1(t), U_2(t), \cdots, U_m(t)\}$ and state data $\{X_1(t), X_2(t), \cdots, X_n(t)\}$ are denoted, respectively, as $u_1(t), u_2(t), \cdots, u_m(t)$ and $x_1(t), x_2(t), \cdots, x_n(t)$. Since the logical matrix S_i for the state variable X_i can be represented by the parameter vector $\boldsymbol{\theta}$, we simply denote them as $S_i(\boldsymbol{\theta})$. Suppose that the state variable X_i can be influenced by the variables $X_{j_1}, X_{j_2}, \cdots, X_{j_k}$. According to (5), it is easy to get expression of the prediction $\hat{x}_i(\boldsymbol{\theta}, t)$:

$$\hat{x}_i(\boldsymbol{\theta}, t) = S_i(\boldsymbol{\theta}) u(t-1) \ltimes_{i=1}^{k} x_{j_i}(t-1) \tag{23}$$

Recalling (21) and (22), the PEM method will estimate the binary parameters by minimizing the prediction error, i.e.,

$$\min_{\boldsymbol{\theta} \in \mathcal{D}^k} \sum_{t=0}^{T} d(X(t), \hat{X}(\boldsymbol{\theta}, t)) = \min_{\boldsymbol{\theta} \in \mathcal{D}^k} \sum_{t=0}^{T} \sum_{i=1}^{n} \left(1 - x_i^{\mathrm{T}}(t) \cdot \hat{x}_i(\boldsymbol{\theta}, t) \right) \tag{24}$$

Furthermore, the optimization problem (24) can be equivalently rewritten as

$$\min_{\boldsymbol{\theta} \in \mathcal{D}^k} \left(T \cdot n - \sum_{t=0}^{T} \sum_{i=1}^{n} x_i^{\mathrm{T}}(t) \cdot \hat{x}_i(\boldsymbol{\theta}, t) \right)$$

which is actually equivalent to

$$\max_{\boldsymbol{\theta} \in \mathcal{D}^k} \sum_{t=0}^{T} \sum_{i=1}^{n} x_i^{\mathrm{T}}(t) \cdot \hat{x}_i(\boldsymbol{\theta}, t) \tag{25}$$

Next, it will be shown that the optimization problem (25) can be formulated as a pseudo-Boolean optimization (i.e., optimization of pseudo-Boolean functions). A pseudo-Boolean function is a mapping from a finite number of Boolean variables to a real number and can be uniquely represented by a multi-linear polynomial (Boros and Hammer, 2002).

As mentioned before, any logical matrix can be expressed by a multi-linear polynomial. After calculation, the term $\sum_{t=0}^{T} \sum_{i=1}^{n} x_i^{\mathrm{T}}(t) \hat{x}_i(\boldsymbol{\theta}, t)$ can be represented by a multivariate polynomial.

$$P_{mv}(\boldsymbol{\theta}) = c + \sum_{\mathcal{Q}_\beta \subset V} c_{\mathcal{Q}_\beta} \prod_{j \in \mathcal{Q}_\beta} \theta_j^{r_{\mathcal{Q}_\beta j}} \tag{26}$$

where $c, c_{\mathcal{Q}_\beta} \in \mathbb{R}$ for $\mathcal{Q}_\beta \subset V = \{1, 2, \cdots, k\}$ and the factor $r_{\mathcal{Q}_\beta, j}, \forall \beta, j$ is a natural number. In addition, using the property of Boolean variables $\theta_i^r = \theta_i, \forall r \in \mathbb{Z}_+$, the multivariate polynomial (26) is easily transformed into a multi-linear polynomial. Consequently, the term $\sum_{t=0}^{T} \sum_{i=1}^{n} x_i^{\mathrm{T}}(t) \cdot \hat{x}_i(\boldsymbol{\theta}, t)$ can be described by a multi-linear polynomial (6) and the optimization problem (25) is transformed into a pseudo-Boolean optimization problem

$$\max_{\boldsymbol{\theta} \in \mathcal{D}^k} P_{ml}(\boldsymbol{\theta}) = \max_{\boldsymbol{\theta} \in \mathcal{D}^k} c + \sum_{i=1}^{k} c_i \theta_i + \sum_{\alpha=1}^{q} c_{\mathcal{I}_\alpha} \prod_{j \in \mathcal{I}_\alpha} \theta_j \tag{27}$$

So far, several different ways to handle the nonlinear pseudo-Boolean optimization problems (27) exist, such as reduction to an equivalent linear or quadratic binary programming problem, branch-and-bound method, linear approximations (Boros and Hammer, 2002; Crama and Rodrí-guez-Heck, 2017). As the linear programming relaxation of an integer linear program can be solved efficiently and based on the solution integer solutions can be found, in this paper we consider "linearization", so that nonlinear binary optimization can be reduced to integer linear program (Crama and Rodrí-guez-Heck, 2017). The key is to introduce auxiliary Boolean variables $\boldsymbol{z} = [z_1 \;\; z_2 \;\; \cdots]^{\mathrm{T}}$ to replace the nonlinear monomial $\prod_{j \in \mathcal{I}_\alpha} \theta_j$ in (6) by means of the AND-expression $z_\alpha = \prod_{j \in \mathcal{I}_\alpha} \theta_j$. Simultaneously to satisfy the AND-expression, linear inequalities as constraints are considered

to get feasible value of the nonlinear monomial $\prod_{j\in\mathcal{I}_\alpha}\theta_j$. Finally, an optimization problem equivalent to (27) is obtained as

$$\begin{aligned}\max_{\boldsymbol{\theta},\boldsymbol{z}} L_P(\boldsymbol{\theta},\boldsymbol{z}) &= \max_{\boldsymbol{\theta},\boldsymbol{z}} c + \sum_{i=1}^{k} c_i\theta_i + \sum_{\alpha} c_{\mathcal{I}_\alpha} z_\alpha \\ s.t. \quad & z_\alpha \leq \theta_j, \forall j \in \mathcal{I}_\alpha, \\ & z_\alpha \geq \sum_{j\in\mathcal{I}_\alpha}\theta_j - (|\mathcal{I}_\alpha| - 1), \\ & z_\alpha \in \mathcal{D},\ \boldsymbol{\theta} \in \mathcal{D}^k.\end{aligned} \tag{28}$$

The constraints in the optimization problem in (27) can be complemented by additional constraints representing the prior knowledge of alternative hypotheses or unateness as shown in Section 4.1 and Section 4.3, respectively.

Remark 2. *It is important to note that minimizing or maximizing a pseudo-Boolean function is known to be $\mathcal{NP}$-hard (Crama and Rodrí-guez-Heck, 2017). However, Breindl et al. (2013) shows that the optimization problem (28) can be solved using a relaxed problem, i.e., linear programming solver based on the simplex method, which requires less computational effort than mixed integer linear program. The relaxed problem delivers an integer as optimal solution, which is also an optimal solution of the optimization problem (28).*

5.2. Handling of Large Scale Networks

With modern measurement techniques it is possible to quantify a huge amount of substances simultaneously. A Boolean network which describes the observed interactions is then also of large scale. But the number of substances which are direct relevant for the regulation of certain substance is usually limited, in other words the connectivity inside the network is bounded. For instance, as pointed out by Arnone and Davidson (1997), the connectivity is bounded by 8. Without prior knowledge the complexity of the algorithm is $\mathcal{O} = 2^{n+m}$ as all state and input combinations have to be considered as potential regulators for all states, even though only some of them are relevant in the end. This would limit the applicability of the approach to rather small networks. If one has hypotheses about potential interaction partners and the number of potential regulators per state is limited by a set of k variables, then the complexity of the algorithm is $\mathcal{O} = 2^k$, as the regulative functions for each state can be inferred separately. The hypotheses for the interaction partners are not necessarily based on prior-knowledge, but could also be computed based on the data set. In Margolin et al. (2006) an approach is presented, which is based on the information theoretic concept of mutual information ranking and the restriction to pairwise interactions that leads to a very good scaling with big data sets.

5.3. Handling of Missing Measurement Values

Dependent on the measurement technique it is sometimes not possible to measure all states at all time instances and the missing values must be handled in the data analysis. There are approaches in the literature to compute an imputation e.g., for microarrays in Gan et al. (2006) and gel-based proteomics in Albrecht et al. (2010). These approaches are based on interpolation or heuristics. An alternative is to use a data analysis approach which can deal with incomplete data matrices.

A missing measurement value can be estimated during the identification by adding additional binary parameters in the identification process. Because of vector expression of states, all possible states belong to the set Δ_{2^n}. In this way, n binary parameters are enough for vector expression of a completely unknown state at time k. For example, if $n = 2$, then we can generally express the unknown state as

$$x(k) = \begin{bmatrix} \gamma_1 \cdot \gamma_2 \\ \gamma_1 \cdot (1-\gamma_2) \\ (1-\gamma_1) \cdot \gamma_2 \\ (1-\gamma_1) \cdot (1-\gamma_2) \end{bmatrix}. \tag{29}$$

Furthermore, as the states of the system are known partially, then the number of binary parameters can be reduced accordingly. So for each missing value one parameter is added to the optimization and the imputation for this value is calculated which fits best to the other dynamic behavior of the system.

5.4. Handling of Unmeasurable Processes

In some systems post transcriptional protein-protein interactions induce dependencies between the measured abundances similar to the transcriptional regulation. This leads to the situation that the transcriptional regulation can not be observed directly and the identification procedure needs to be adapted accordingly (Geier et al., 2007). The dependencies between the states and the measured outputs can be included in boolean models easily by adding Boolean functions mapping from the actual stats $X(t)$ to the measured outputs $Y(t)$:

$$Y_j(t) = h_j(X(t)), \qquad j = 1, 2, \ldots, p \tag{30}$$

where $[Y(t) = Y_1(t)\ Y_2(t)\ \ldots\ Y_p(t)]^T \in \mathcal{D}^p$ is the output vector at time t , h_i are logic functions. All structural information on the logic functions can be expressed with a logical matrix H

$$y(t) = Hx(t) \tag{31}$$

which can be derived analogous to Equations (2–5). All approaches presented in this paper can be extended for the BN model with output mapping. As additional logic functions are to be identified, additional unknown parameters are added and these parameters cannot be separately identified from the parameters of the regulative functions, which impacts the computational burden drastically (Zhang et al., 2017b) .

5.5. Influence of Noise

In real world experiments measurement noise is unavoidable. With a sophisticated binarization method the influence of additive noise can often be suppressed (Hopfensitz et al., 2012). But noise can still lead to wrong binarized values in some cases and consequently errors in the input to the identification method

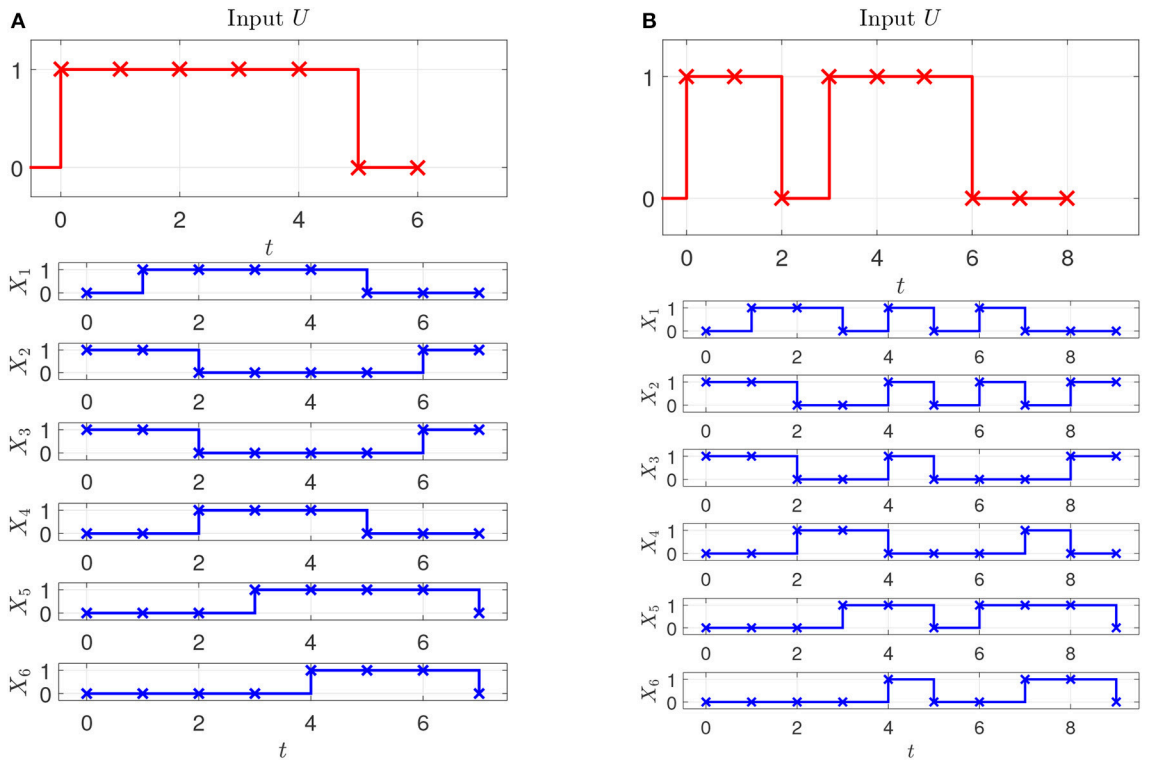


FIGURE 2 | Perturbation and state measurement. **(A)** First experiment. **(B)** Second experiment.

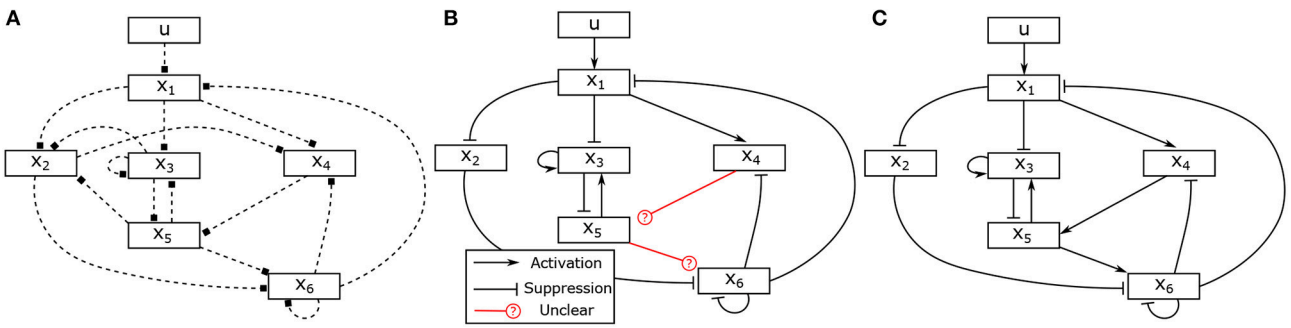


FIGURE 3 | Hypothesis, partially identified and fully identified network graph. **(A)** Hypotheses for regulative interactions. **(B)** Identified Boolean network without canalizing information. **(C)** Identified Boolean network with prior knowlege.

cannot be totally avoided. As the presented approach is based on an optimization, the network which optimally fits to the observed data is found. Inconsistent transitions caused by noise in the data set can be handled directly and lead to an identification result with a non-zero prediction error. If, due to noise, the observed transitions would lead to an identification result which is contradictory to prior knowledge, the identification approach ignores these transitions directly.

Example 6. Consider the BCN for oxidative stress response pathways with the PI3-Kinase-Akt pathway given in Sridharan et al. (2012).

$$\begin{cases} X_1(t+1) &= U(t) \wedge \neg X_6(t) \\ X_2(t+1) &= \neg X_1(t) \\ X_3(t+1) &= \neg X_1(t) \wedge (X_5(t) \vee X_3(t) \\ X_4(t+1) &= X_1(t) \wedge \neg X_6(t) \\ X_5(t+1) &= X_4(t) \vee \neg X_3(t) \\ X_6(t+1) &= X_5(t) \wedge (\neg X_6(t) \vee \neg X_2(t)) \end{cases} \tag{32}$$

In the model, X_1 represents stress reactive intermediaries, X_2 transcription factor A, X_3 key protein, X_4 protein kinase, X_5

transcription factor B, X_6 anti-stress response element, U stress signal. Using STP, (32) can be converted into the algebraic form (5) with $x(t) = \ltimes_{i=1}^{6} x_i(t) \in \Delta_{64}, u(t) \in \Delta_2$.

Assume that two experiments have been executed starting in steady state with two different stimuli, the corresponding input-state data is obtained as shown in **Figures 2A,B**. Assume further that as prior knowledge the candidates of regulative interactions (see the dashed lines in **Figure 3A**) and the attractor are given. The attractor of the BCN without stress is $X_1 = 0,\ X_2 = 1,\ X_3 = 1, X_4 = 0,\ X_5 = 0,\ X_6 = 0$.

Based on the candidates of regulative interactions, the number of unknown binary parameters $\boldsymbol{\theta}$ representing the logical matrices of the Boolean functions can be reduced from $6 \cdot 2^7 = 768$ to 40 as described in Section 4.1. For instance, since the variable X_2 is connected with the variables X_1, X_3 and X_5, it means that the Boolean function of the variable X_2 can be described by $f_2(X_1, X_5, X_5)$. Accordingly, 8 binary parameters are enough to represent the logical matrix S_2 of the Boolean function f_2, i.e.,

$$S_2 = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 & \theta_7 & \theta_8 \\ 1-\theta_1 & 1-\theta_2 & 1-\theta_3 & 1-\theta_4 & 1-\theta_5 & 1-\theta_6 & 1-\theta_7 & 1-\theta_8 \end{bmatrix}. \tag{33}$$

The information about the steady state is used as described in Section 4.4 to determine one parameter in each matrix, which reduces the number of unknown variables to 34. In the next, we apply the proposed approach to identify the model of the BCN from the given input-state data. Solving the optimization problem (28), in total, 31 unknown binary parameters can be determined. The identification result is depicted in **Figure 3B** and the identified matrices are as follows,

$$\begin{aligned} S_1 &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}, \qquad S_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \\ S_3 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \\ S_4 &= \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}, \ S_5 = \begin{bmatrix} \theta_{29} & 0 & 1 & 1 \\ 1-\theta_{29} & 1 & 0 & 0 \end{bmatrix}, \\ S_6 &= \begin{bmatrix} 0 & 1 & \theta_{35} & 0 & 1 & 1 & \theta_{39} & 0 \\ 1 & 0 & 1-\theta_{35} & 1 & 0 & 0 & 1-\theta_{39} & 1 \end{bmatrix}. \end{aligned} \tag{34}$$

It can be seen that the logical matrices of the Boolean functions for X_5 and X_6 can not be uniquely determined. Combined with an additional information about activating or suppressing properties of the states, for instance, X_4 and X_5 are, respectively, activator to X_5 and X_6, the complete model can be uniquely reconstructed. The canalizing property of X_4 and X_5 can be utilized as described in Section 4.2. If this information is not available, one could conduct additional experiments with different stimuli and combine the data to have full reconstruction of the model as depicted in **Figure 3C**.

6. DISCUSSION

The proposed method facilitates the incorporation of various types of prior knowledge. The optimization problem can be solved by efficient linear programming solvers. By using the simplex method one can guarantee to find the network which optimally fits to the observed data. In comparison, the genetic algorithms based approaches may not guarantee the optimal solution. The proposed method is developed for synchronous Boolean networks. It can be applied to large scale networks, if the connectivity of the network to be identified is limited with aid of prior knowledge or application of information theory.

In future we plan to investigate data-based approaches to infer the connections in large networks and automated partitioning into smaller subsystems (e.g., with an adapted approach from discrete event systems like Saives et al., 2018). We also work on a new method for the binarization based on the idea that the qualitative system behavior before and after the binarization shall be the same.

AUTHOR CONTRIBUTIONS

TL, ZZ, and PZ conception and design of research. TL and ZZ performed simulation and analyzed data. TL, ZZ, and PZ interpreted simulation results. TL and ZZ prepared figures. TL, ZZ, and PZ drafted manuscript and approved final version of manuscript.

FUNDING

This work is supported by the Federal State of Rhineland-Palatinate, Germany in the framework of the project Complex Data Analysis in Life Sciences and Biotechnology (*BioComp*).

ACKNOWLEDGMENTS

The authors would like to thank Michael Schroda and Timo Mühlhaus for the discussion on the data characteristics in biological systems.

REFERENCES

Akutsu, T., Miyano, S., and Kuhara, S. (1999). "Identification of genetic networks from a small number of gene expression patterns under the boolean network model," in *Proceedings of the Pacific Symposium on Biocomputing* (Mauna Lani), 17–28.

Albert, R., and Barabási, A.-L. (2000). Dynamics of complex systems: Scaling laws for the period of boolean networks. *Phys. Rev. Lett.* 84, 5660–5663. doi: 10.1103/PhysRevLett.84.5660

Albrecht, D., Kniemeyer, O., Brakhage, A. A., and Guthke, R. (2010). Missing values in gel-based proteomics. *Proteomics* 10, 1202–1211. doi: 10.1002/pmic.200800576

Alon, N., Babai, L., and Suzuki, H. (1991). Multilinear polynomials and frankl-ray-chaudhuri-wilson type intersection theorems. *J. Combinat. Theory A* 58, 165–180. doi: 10.1016/0097-3165(91)90058-O

Arnone, M., and Davidson, E. (1997). The hardwiring of development: organization and function of genomic regulatory systems. *Development* 124, 1851–1864.

Berestovsky, N., and Nakhleh, L. (2013). An evaluation of methods for inferring boolean networks from time-series data. *PLoS ONE* 8:e66031. doi: 10.1371/journal.pone.0066031

Bollobas, B. (2012). *Graph Theory: An Introductory Course*, Vol. 63. New York, NY: Springer Science & Business Media.

Boros, E., and Hammer, P. L. (2002). Pseudo-boolean optimization. *Discrete Appl. Math.* 123, 155–225. doi: 10.1016/S0166-218X(01)00341-9

Breindl, C., Chaves, M., and Allgöwer, F. (2013). "A linear reformulation of boolean optimization problems and structure identification of gene regulation networks," in *Proceedings of the 52th IEEE Conference on Decision and Control* (Florence), 733–738.

Cheng, D. (2001). Semi-tensor product of matrices and its application to morgen's problem. *Sci. China Ser. Informat. Sci.* 2001, 195–212. doi: 10.1007/BF02714570

Cheng, D., and Qi, H. (2010). A linear representation of dynamics of boolean networks. *IEEE Trans. Automat. Cont.* 55, 2251–2258. doi: 10.1109/TAC.2010.2043294

Cheng, D., Qi, H., and Li, Z. (2011a). *Analysis and Control of Boolean Networks: A Semi-Tensor Product Approach*. London: Springer.

Cheng, D., Qi, H., and Li, Z. (2011b). Model construction of boolean network via observed data. *IEEE Trans. Neural Netw.* 22, 525–536. doi: 10.1109/TNN.2011.2106512

Cheng, D., and Zhao, Y. (2011). Identification of boolean control networks. *Automatica* 47, 702–710. doi: 10.1016/j.automatica.2011.01.083

Crama, Y., and Rodrí-guez-Heck, E. (2017). A class of valid inequalities for multilinear 0-1 optimization problems. *Discrete Optimizat.* 25, 28–47. doi: 10.1016/j.disopt.2017.02.001

Davidich, M. I., and Bornholdt, S. (2008). Boolean network model predicts cell cycle sequence of fission yeast. *PLoS ONE* 3:e1672. doi: 10.1371/journal.pone.0001672

Dorier, J., Crespo, I., Niknejad, A., Liechti, R., Ebeling, M., and Xenarios, I. (2016). Boolean regulatory network reconstruction using literature based knowledge with a genetic algorithm optimization method. *BMC Bioinform.* 17:410. doi: 10.1186/s12859-016-1287-z

Faisal, S., Lichtenberg, G., Trump, S., and Attinger, S. (2010). Structural properties of continuous representations of boolean functions for gene network modelling. *Automatica* 46, 2047–2052. doi: 10.1016/j.automatica.2010.09.001

Fauré, A., Naldi, A., Chaouiya, C., and Thieffry, D. (2006). Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle. *Bioinformatics* 22, e124–e131. doi: 10.1093/bioinformatics/btl210

Fornasini, E., and Valcher, M. E. (2014). "Identification problems for boolean networks and boolean control networks," in *Proceedings of the 19th IFAC World Congress* (Cape Town), 5399–5404.

Fumia, H. F., and Martins, M. L. (2013). Boolean network model for cancer pathways: predicting carcinogenesis and targeted therapy outcomes. *PLoS ONE* 8:e69008. doi: 10.1371/journal.pone.0069008

Gan, X., Liew, A. W.-C., and Yan, H. (2006). Microarray missing data imputation based on a set theoretic framework and biological knowledge. *Nucleic Acids Res.* 34, 1608–1619. doi: 10.1093/nar/gkl047

Geier, F., Timmer, J., and Fleck, C. (2007). Reconstructing gene-regulatory networks from time series, knock-out data, and prior knowledge. *BMC Sys. Biol.* 1:11. doi: 10.1186/1752-0509-1-11

Grieb, M., Burkovski, A., Sträng, J. E., Kraus, J. M., Groß, A., Palm, G., et al. (2015). Predicting variabilities in cardiac gene expression with a boolean network incorporating uncertainty. *PLoS ONE* 10:e0131832. doi: 10.1371/journal.pone.0131832

Hamming, R. W. (1950). Error detecting and error correcting codes. *Bell Labs Techn. J.* 29, 147–160. doi: 10.1002/j.1538-7305.1950.tb00463.x

Higa, C. H., Louzada, V. H., Andrade, T. P., and Hashimoto, R. F. (2011). Constraint-based analysis of gene interactions using restricted boolean networks and time-series data. *BMC Proc.* 5:S5. doi: 10.1186/1753-6561-5-S2-S5

Hopfensitz, M., Müssel, C., Wawra, C., Maucher, M., Kühl, M., Neumann, H., and Kestler, H. A. (2012). Multiscale binarization of gene expression data for reconstructing boolean networks. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 9, 487–498. doi: 10.1109/TCBB.2011.62

Huang, S., and Ingber, D. E. (2000). Shape-dependent control of cell growth, differentiation, and apoptosis: switching between attractors in cell regulatory networks. *Exp. Cell Res.* 261, 91–103. doi: 10.1006/excr.2000.5044

Isermann, R., and Münchhof, M. (2011). *Identification of Dynamic Systems: An Introduction With Applications*. Berlin;Heidelberg: Springer.

Karlebach, G., and Shamir, R. (2012). Constructing logical models of gene regulatory networks by integrating transcription factor-dna interactions with expression data: an entropy-based approach. *J. Comput. Biol.* 19, 30–41. doi: 10.1089/cmb.2011.0100

Kauffman, S. (1974). The large scale structure and dynamics of gene control circuits: an ensemble approach. *J. Theor. Biol.* 44, 167–190. doi: 10.1016/S0022-5193(74)80037-8

Kauffman, S., Peterson, C., Samuelsson, B., and Troein, C. (2003). Random boolean network models and the yeast transcriptional network. *Proc. Natl. Acad. Sci. U.S.A.* 100, 14796–14799. doi: 10.1073/pnas.2036429100

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.* 22, 437–467. doi: 10.1016/0022-5193(69)90015-0

Lähdesmäki, H., Shmulevich, I., and Yli-Harja, O. (2003). On learning gene regulatory networks under the boolean network model. *Mach. Learn.* 52, 147–167. doi: 10.1023/A:1023905711304

Liang, S., Fuhrman, S., and Somogyi, R. (1998). "Reveal: a general reverse engineering algorithm for inference of genetic network architectures," in *Proceedings of the Pacific Symposium on Biocomputing* (Hawaii), 18–29.

Margolin, A. A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R. D., et al. (2006). Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformat.* 7:S7. doi: 10.1186/1471-2105-7-S1-S7

Naldi, A., Monteiro, P. T., Müssel, C., Consortium for Logical Models and Tools, Kestler, H. A., Thieffry, D., et al. (2015). Cooperative development of logical modelling standards and tools with colomoto. *Bioinformatics* 31, 1154–1159. doi: 10.1093/bioinformatics/btv013

Ostrowski, M., Paulevé, L., Schaub, T., Siegel, A., and Guziolowski, C. (2016). Boolean network identification from perturbation time series data combining dynamics abstraction and logic programming. *Biosystems* 149, 139–153. doi: 10.1016/j.biosystems.2016.07.009

Ouyang, H., Fang, J., Shen, L., Dougherty, E. R., and Liu, W. (2014). Learning restricted boolean network model by time-series data. *EURASIP J. Bioinform. Sys. Biol.* 2014:10. doi: 10.1186/s13637-014-0010-5

Porreca, R., Cinquemani, E., Lygeros, J., and Ferrari-Trecate, G. (2010). Identification of genetic network dynamics with unate structure. *Bioinformatics* 26, 1239–1245. doi: 10.1093/bioinformatics/btq120

Saives, J., Faraut, G., and Lesage, J. J. (2018). Automated partitioning of concurrent discrete-event systems for distributed behavioral identification. *IEEE Trans. Autom Sci. Eng.* 15, 832–841. doi: 10.1109/TASE.2017.2718244

Sridharan, S., Layek, R., Datta, A., and Venkatraj, J. (2012). Boolean modeling and fault diagnosis in oxidative stress response. *BMC Genomics* 13(Suppl. 6):S4. doi: 10.1186/1471-2164-13-S6-S4

Terfve, C., Cokelaer, T., Henriques, D., MacNamara, A., Goncalves, E., Morris, M. K., et al. (2012). Cellnoptr: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms. *BMC Sys. Biol.* 6:133. doi: 10.1186/1752-0509-6-133

Videla, S., Guziolowski, C., Eduati, F., Thiele, S., Gebser, M., Nicolas, J., et al. (2015). Learning boolean logic models of signaling networks with asp. *Theor. Comput. Sci.* 599, 79–101. doi: 10.1016/j.tcs.2014.06.022

Waddington, C. H. (1942). Canalization of development and the inheritance of acquired characters. *Nature* 150, 563–565. doi: 10.1038/150563a0

Wang, R.-S., Saadatpour, A., and Albert, R. (2012). Boolean modeling in systems biology: an overview of methodology and applications. *Phys. Biol.* 9:055001. doi: 10.1088/1478-3975/9/5/055001

Zhang, X., Han, H., and Zhang, W. (2017a). Identification of boolean networks using premined network topology information. *IEEE Trans. Neural Netw. Learn. Sys.* 28, 464–469. doi: 10.1109/TNNLS.2016.2514841

Zhang, Z., Leifeld, T., and Zhang, P. (2017b). "Identification of boolean control networks incorporating prior knowledge," in *IEEE 56th Annual Conference on Decision and Control* (Melbourne, VIC), 5839–5844.

Zhou, X., Wang, X., and Dougherty, E. R. (2003). Binarization of microarray data on the basis of a mixture model. *Mol. Cancer Therapeut.* 2, 679–684.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

PROTOCOLS
published: 19 June 2018
doi: 10.3389/fphys.2018.00646

Logical Modeling and Analysis of Cellular Regulatory Networks With GINsim 3.0

Aurélien Naldi[1], Céline Hernandez[1], Wassim Abou-Jaoudé[1], Pedro T. Monteiro[2], Claudine Chaouiya[3]* and Denis Thieffry[1]**

[1] Computational Systems Biology Team, Institut de Biologie de l'Ecole Normale Supérieure (IBENS), École Normale Supérieure, Centre National de la Recherche Scientifique, Institut National de la Sante et de la Recherche Médicale, PSL Université, Paris, France, [2] INESC-ID, Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal, [3] Instituto Gulbenkian de Ciência, Oeiras, Portugal

The logical formalism is well adapted to model large cellular networks, in particular when detailed kinetic data are scarce. This tutorial focuses on this well-established qualitative framework. Relying on GINsim (release 3.0), a software implementing this formalism, we guide the reader step by step toward the definition, the analysis and the simulation of a four-node model of the mammalian p53-Mdm2 network.

Keywords: regulatory network, logical model, discrete dynamics, regulatory circuit, p53-Mdm2 network

OPEN ACCESS

Edited by:
Theodore J. Perkins, University of Ottawa, Canada

Reviewed by:
Elena S. Dimitrova, Clemson University, United States
Jim Rogers, University of Nebraska Omaha, United States

****Correspondence:***
Aurélien Naldi
aurelien.naldi@ens.fr
Claudine Chaouiya
chaouiya@igc.gulbenkian.pt
Denis Thieffry
thieffry@ens.fr

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: *05 March 2018*
Accepted: *11 May 2018*
Published: *19 June 2018*

Citation:
Naldi A, Hernandez C, Abou-Jaoudé W, Monteiro PT, Chaouiya C and Thieffry D (2018) Logical Modeling and Analysis of Cellular Regulatory Networks With GINsim 3.0. Front. Physiol. 9:646. doi: 10.3389/fphys.2018.00646

1. INTRODUCTION

The logical formalism is becoming increasingly popular to model cellular networks (Naldi et al., 2015; Abou-Jaoudé et al., 2016). Here, we focus on the framework developed by René Thomas and colleagues, which includes the use of multi-valued variables when functionally justified, along with sophisticated logical rules or parameters (Thomas, 1991; Thomas et al., 1995).

This approach has been applied to the study of a wide range of networks controlling, for example, the lysis-lysogeny decision of the bacteriophage λ (Thieffry and Thomas, 1995), the specification of flower organs in arabidopsis (Mendoza et al., 1999; Azpeitia et al., 2014), the segmentation of drosophila embryo (Sánchez and Thieffry, 2001; Sánchez and Thieffry, 2003; Sánchez et al., 2008; Mbodj et al., 2016), the specification of compartments in drosophila imaginal disks (González et al., 2006, 2008), drosophila egg shell patterning (Fauré et al., 2014), the control of cell cycle in yeast and mammals (Fauré et al., 2006, 2009; Traynard et al., 2016), the specification of immune cells from common progenitors (Mendoza and Méndez, 2015; Collombet et al., 2017), the differentiation of T-helper lymphocytes (Naldi et al., 2010; Abou-Jaoudé et al., 2015; Martinez-Sanchez et al., 2015), neuronal differentiation (Coolen et al., 2012), as well as cancer cell fate decisions (Sahin et al., 2009; Calzone et al., 2010; Grieco et al., 2013; Flobak et al., 2015; Remy et al., 2015), etc.

In order to ease access to logical modeling by biologists, this protocol proposes a stepwise introduction to the framework, relying on its implementation into the software GINsim (release 3.0). The following section introduces the biological system used as an illustration. Next, in section 3, we proceed with the stepwise construction and analysis of a logical model. Section 4 covers potential troubleshooting. The article then ends with some conclusions and prospects.

2. THE P53-MDM2 NETWORK

The transcription factor p53 plays an essential role in the control of cell proliferation in mammals by regulating a large number of genes involved notably in growth arrest, DNA repair, or apoptosis

(Vogelstein et al., 2000). Its level is tightly regulated by the ubiquitin ligase Mdm2. More precisely, nuclear Mdm2 down-regulates the level of active p53, both by accelerating p53 degradation through ubiquitination (Brooks and Gu, 2006) and by blocking the transcriptional activity of p53 (Oliner et al., 1993; Coutts et al., 2007). In turn, p53 activates Mdm2 transcription (Barak et al., 1993) and down-regulates the level of nuclear Mdm2 by inhibiting Mdm2 nuclear translocation through inactivation of the kinase Akt (Mayo and Donner, 2002). Finally, high levels of p53 promote damage repair by inducing the synthesis of DNA repair proteins (Gatz and Wiesmüller, 2006).

Given its key role in DNA repair and cell fate control, various groups have modeled this network using different formalisms, including ordinary differential equations (Ciliberto et al., 2005; Zhang et al., 2011), stochastic models (Puszynski et al., 2008; Ouattara et al., 2010; Sun and Cui, 2014), hybrid deterministic and stochastic models (Iwamoto et al., 2014), as well as logical models (Abou-Jaoudé et al., 2009; Choi et al., 2012).

In this protocol, we rely on a refined version of a logical model presented by Abou-Jaoudé et al. (2009), involving the protein p53, the ubiquitin ligase Mdm2 in the cytoplasm, the ubiquitin ligase Mdm2 in the nucleus, and DNA damage (see **Figure 1**).

3. CONSTRUCTION AND ANALYSIS OF THE MODEL

In this section, referring to the p53-Mdm2 network defined above, we introduce the different steps required for the definition of a logical model and for the analysis of its dynamical properties with the software GINsim, release 3.0.

3.1. GINsim

The GINsim software supports the definition, the simulation and the analysis of regulatory graphs, based on the (multi-valued) logical formalism. GINsim is freely available from its dedicated website (http://ginsim.org), along with documentation and a model repository. For this tutorial, we use the recent release 3.0, which is available for all platforms with version 8 of the Java Virtual Machine.

To get started with GINsim, download the corresponding Java ARchive (JAR file), with dependencies included, from the download section of GINsim website (http://ginsim.org/downloads). On your computer, double-click on the file icon to start the application or launch it with the command: `java -jar GINsim-#version.jar` in a terminal. Further instructions, troubleshooting and options are documented on the website.

3.2. Definition of a Logical Regulatory Graph

Upon launch, GINsim displays a window enabling the creation of a new model, the import of a model in a supported format, or the opening of a previously defined model (if any). By clicking on the *New model* button, a window enabling the edition of a new logical regulatory graph opens.

To edit a graph, use the toolbox located just on the top of the window (below the menu bar, see **Figure 2**). Passing slowly

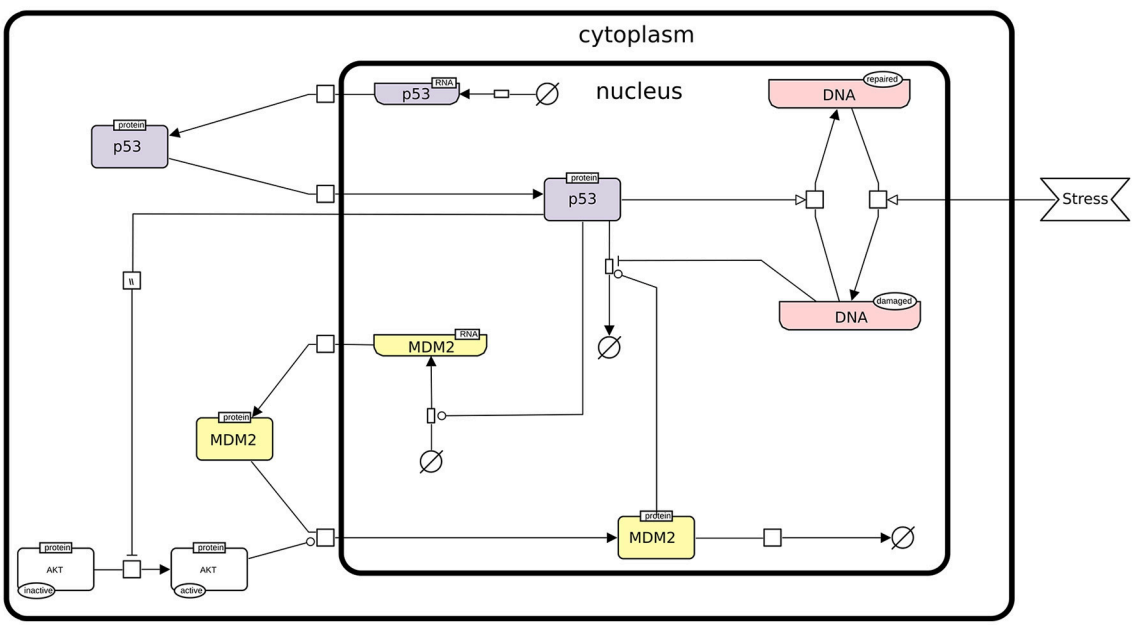


FIGURE 1 | The p53-Mdm2 network. This figure describes the interactions between p53, Mdm2, and DNA damage. An external stress induces a damage to the DNA, which promotes Mdm2 degradation. The level of p53 can then increase and activate DNA repair mechanisms. In parallel, p53 inhibits Mdm2 translocation from the cytoplasm to the nucleus through the inactivation of AKT. However, in the nucleus, high level of p53 activates Mdm2 transcription, while Mdm2 induces the degradation of p53, thereby forming a negative feedback circuit. This figure has been drawn according to the Systems Biology Graphical Notation (SBGN) specifications (Le Novère et al., 2009).

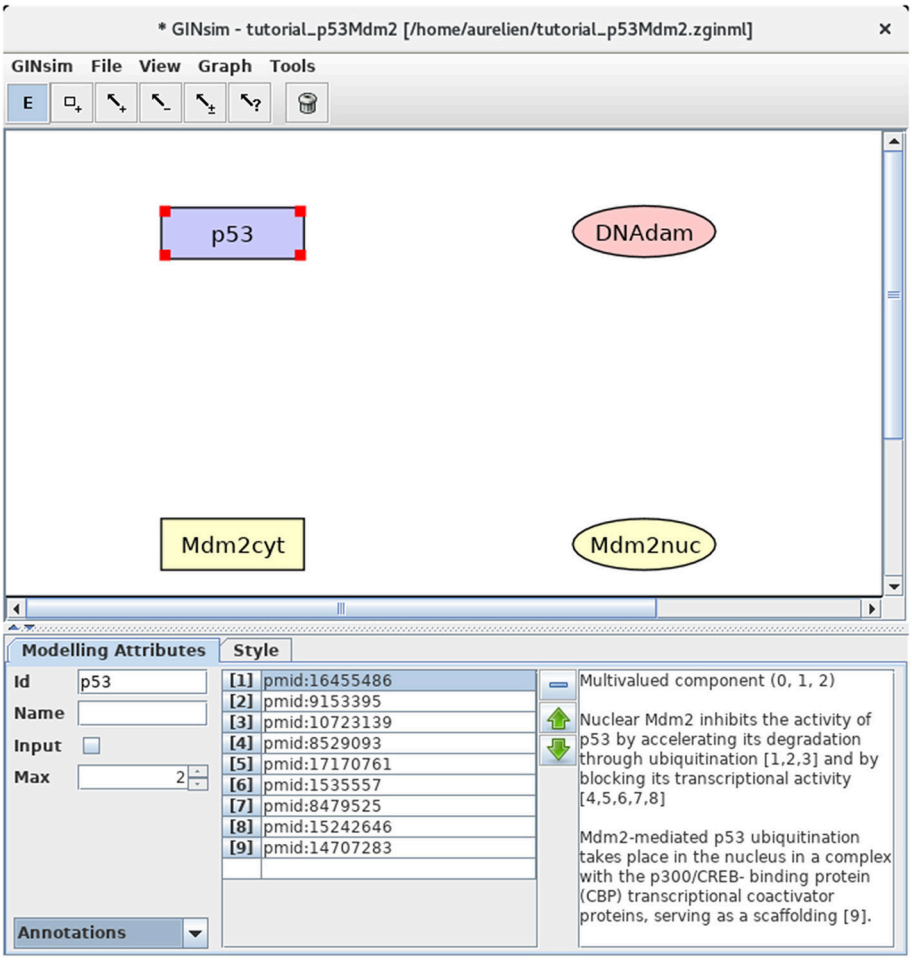


FIGURE 2 | GINsim main window displaying the nodes of the p53-Mdm2 logical regulatory graph. The upper part of the window displays five scrolling menus. These menus provide access to classical file management options, as well as exports into various formats. The central area displays the regulatory graph (here the nodes of the p53-Mdm2 model), while the other area contains two tabs: the *Modeling Attributes* tab (selected here) and the *Style* tab, corresponding to the selected node, here p53. The graphical appearance of the nodes have been modified using the *Style* tab. The *Edit* button on the top is selected and emphasized in blue, enabling the edition of the attributes of the selected node, including its id and name, its maximal level (*Max*, here set to 2), and also the insertion of annotations in the form of free text (bottom right) or of links to relevant database entries (bottom middle).

with the mouse on each of the editing tools displays a message explaining the function of each tool. Clicking on the *E* icon enables further edition of an existing node or arc upon selection, while the garbage can icon serves to delete selected arcs and nodes. Clicking once on one of the remaining icons enables the drawing of a single node or arc. Clicking twice on one of these tools locks the corresponding editing mode, enabling the drawing of several nodes or arcs without clicking repeatedly on the same tool.

TABLE 1 | Regulatory nodes and maximal levels for the p53-Mdm2 model.

Regulatory nodes	Maximal levels
p53	2
Mdm2cyt	2
Mdm2nuc	1
DNAdam	1

3.2.1. Definition of the Regulatory Nodes

First, we need to define four nodes for the four key regulatory factors of the model: p53, Mdm2cyt, Mdm2nuc, and DNA damage (DNAdam). Each node has a unique identifier and a maximal level, specifying a range of possible functional qualitative levels, as listed in **Table 1**. To define all the nodes in a row, first double-click on the node addition tool (symbol is a square with a plus sign) to lock this mode, then click four times on the panel to create the four nodes, with default identifiers and a maximal level of 1. Next, click on the *E* icon to stop adding nodes, and select each node to change its ID and maximal level (when required) in the bottom edition panel. **Figure 2** illustrates this step.

3.2.2. Definition of Regulatory Interactions

Next, we need to define the arcs representing the regulatory interactions between the factors considered in the model. An arc is defined by its source and target nodes, a sign, and a threshold, as described in **Table 2** and illustrated in **Figure 3**. In the non-Boolean case, a node may have distinct actions on a target node, depending on its activity level (e.g., from Mdm2cyt onto Mdm2nuc). In this case, one arc is drawn, which encompasses multiple interactions, each with its own threshold. An interaction is then active when the level of its source is equal or above its threshold, but below the threshold of the next interaction. Add each arc between each relevant pair of nodes by selecting the relevant tool (addition of positive, negative, dual, or unknown interaction) and dragging a line from the source to the target node. Next, use the edition panel to specify multiple interactions with their thresholds, and possibly change their signs.

TABLE 2 | Interactions and thresholds for the p53-Mdm2 model.

Source nodes	Target nodes	Thresholds	Interaction signs
p53	Mdm2nuc	1	–
	Mdm2cyt	2	+
	DNAdam	2	–
Mdm2cyt	Mdm2nuc	1	+
		2	+
Mdm2nuc	p53	1	–
DNAdam	DNAdam	1	+
	Mdm2nuc	1	–

3.2.3. Definition of the Regulatory Rules

We can now define the rules governing the evolution of the regulatory node levels. For each node, specify the logical rules listed in **Table 3**. For this, select a node and the *Formulae* view in the drop-down list at the bottom left of the GINsim window. Click on the little arrow in the main bottom panel, expand the tree view and then click on the *E* button, to enter a formula.

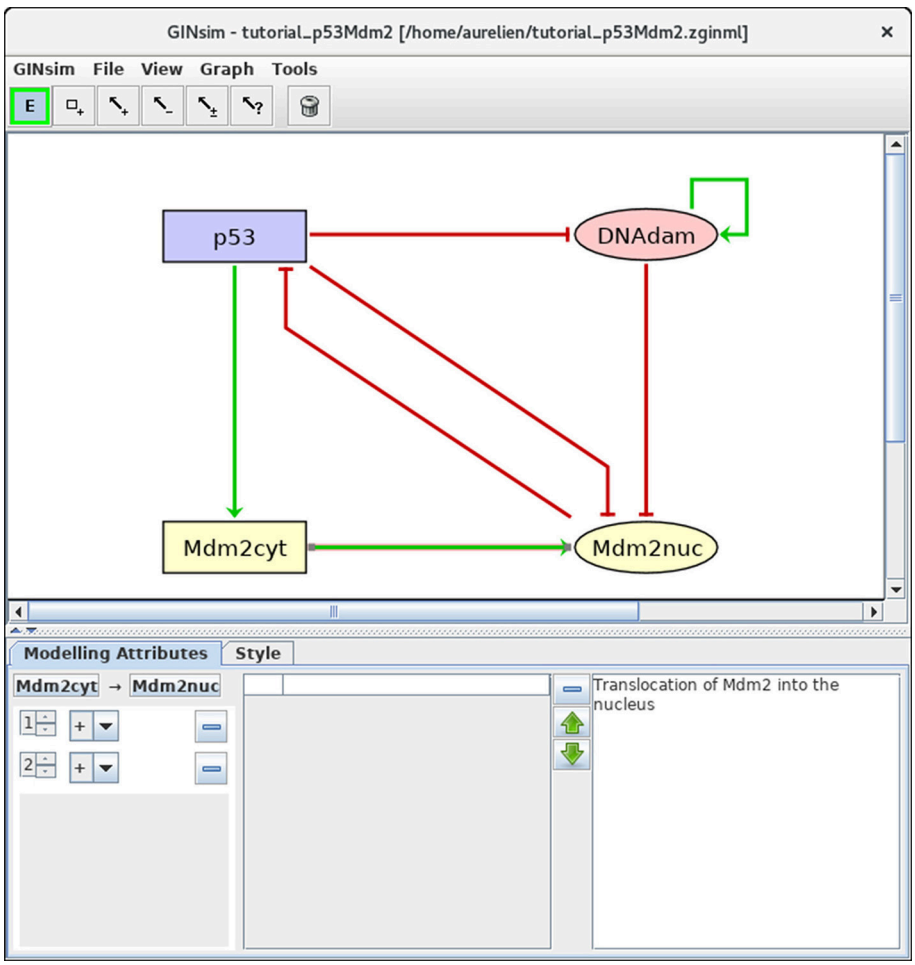


FIGURE 3 | Regulatory arc management in GINsim. To add an arc, the corresponding arc button must be pushed (push twice to add several arcs in one go), allowing the drawing of an arc between a source node and its target. Once an arc has been defined, it can be further edited by selecting it after locking the *E* button. The sign and threshold of the interaction(s) associated with an arc are defined within the *Modeling Attributes* tab, as shown here for the arc from Mdm2cyt onto Mdm2nuc. The additional interaction with threshold level 2 was created by clicking on the + button displayed when additional thresholds are available.

Figure 4 illustrates this step. Note that the definition of adequate logical rules (or parameters, see Note 1) is necessary to ensure the desired effects of each interaction on the target nodes. Per default, GINsim assigns a null target value to each node devoid of explicit rule.

TABLE 3 | Logical rules for the nodes of the p53-Mdm2 model.

Regulatory nodes	Target levels	Boolean rules
p53	2	!Mdm2nuc
Mdm2cyt	2	p53
	1	!p53
Mdm2nuc	1	Mdm2cyt:2 | (Mdm2cyt:1 & !p53 & !DNAdam)
DNAdam	1	DNAdam & !p53

This table lists the conditions enabling the activation of each node (up to level one in the case of a Boolean node, potentially up to higher levels for multi-valued nodes, as for p53 and Mdm2cyt here). These conditions are defined in term of Boolean expressions using the NOT, AND and (inclusive) OR Boolean operators (denoted by !, & and | in GINsim, respectively).

3.2.4. Adding Annotations

To keep track of supporting data and modeling assumptions, the user can add textual annotations and hyperlinks to relevant database entries, at the level of the model itself, as well as for each individual node or arc (see **Figure 2** for an illustration). While the annotation panel is always visible when editing an arc, it requires to select the *Annotations* view (in the bottom left drop-down list) when editing a node.

3.2.5. Changing Layout and Styles

The layout and graphical appearance of nodes and arcs of the graph can be changed according to the user taste. For this, select a node or an arc, along with the *Style* tab. The user can further

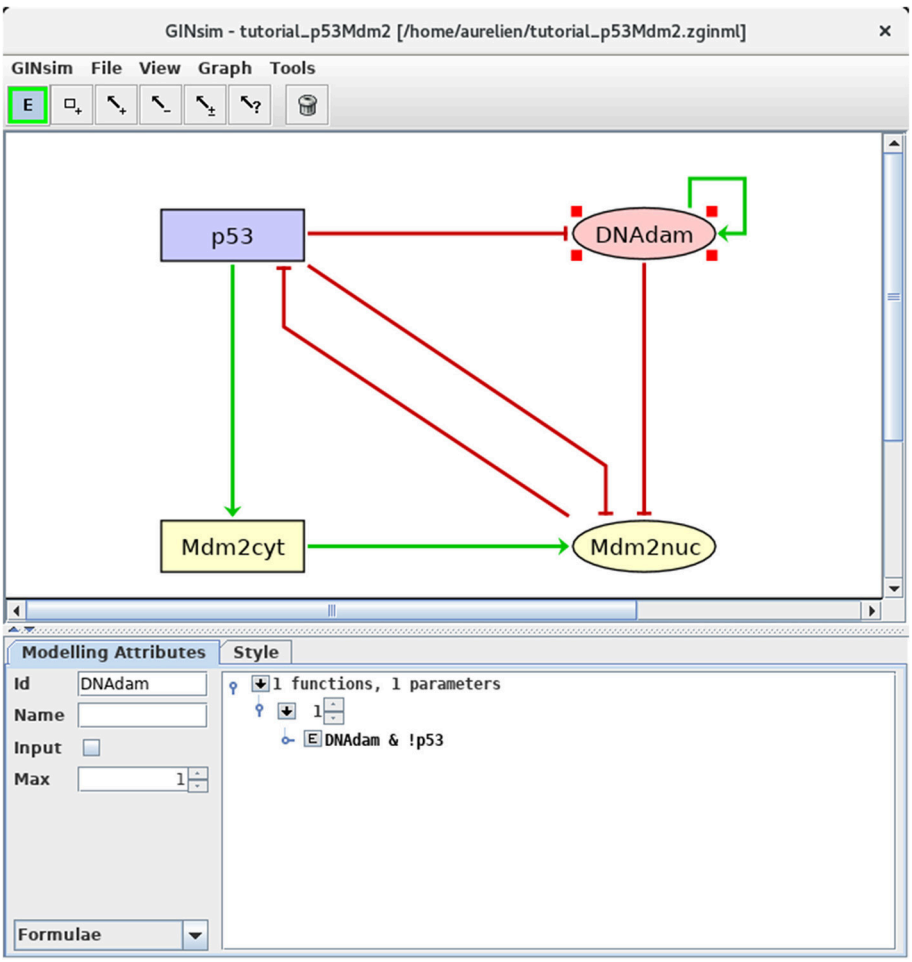


FIGURE 4 | Defining logical rules for the regulatory nodes. This screenshot shows the *Modeling Attributes* associated with the selected node DNAdam. The maximal level is set to 1. After selecting *Formulae* with the bottom-left scrolling menu, the user can enter logical formulae by clicking on the little arrows in the main bottom. The target level (set to 1 per default) can be changed in the case of a multi-valued node. By clicking on the *E* button, one can directly write a formula, using literals (these should exactly match the IDs of nodes regulating the selected node, i.e., p53 or DNAdam in the present case) and the Boolean operators !, & and |, denoting NOT, AND and (inclusive) OR, respectively (following the usual priority ordering; parentheses can be used to define complex formulae). Note that several rows can be used in association with a single target value; these rows are then combined with OR operators. Here, the formula *DNAdam & !p53* associated with the target value 1 implies that DNAdam will be maintained at a level 1 if already present, but only in the absence of p53.

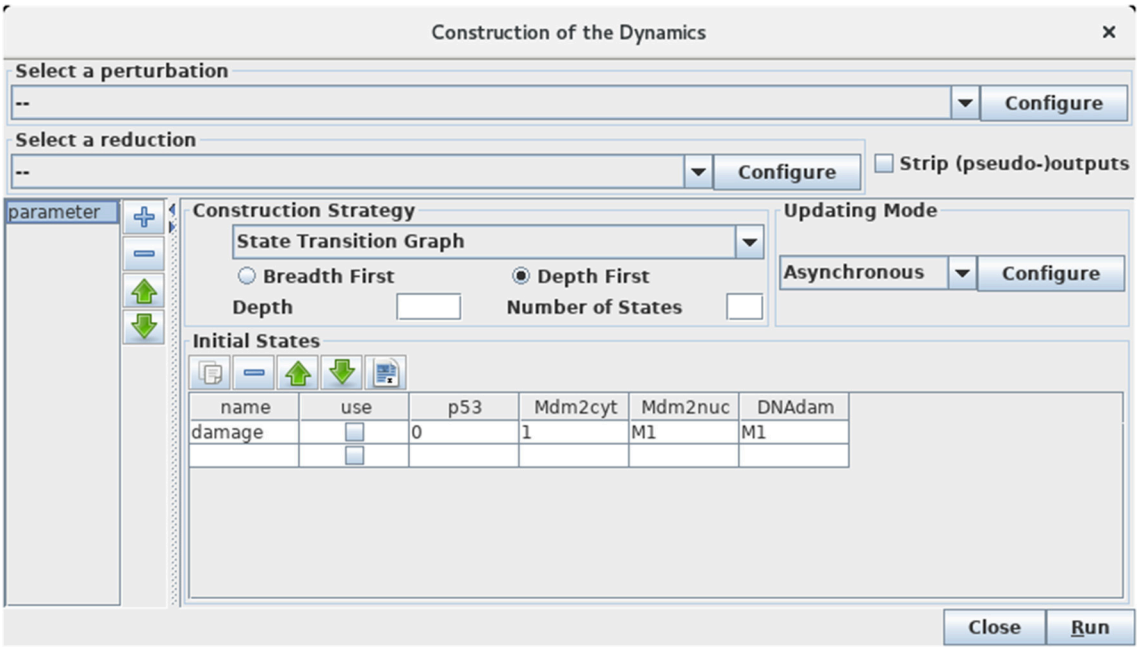


FIGURE 5 | Launching of the construction of a state transition graph. This panel is obtained when selecting *Run Simulation* from the *Tools* scrolling menu in GINsim main window. The default simulation settings are shown, i.e., the construction of a state transition graph using the asynchronous updating, with no specified initial state (meaning that all states are considered in the simulation). Hitting the *Run* button will generate the corresponding state transition graph, which can be displayed in a new window (see **Figure 6**). In the table under *Initial States*, one can define one or several initial states from which the dynamics will be constructed (just type the desired values in a row along with an optional name). Each row of the table defines a single pattern of states, and the check-boxes allow to select the states to be used for a simulation. The levels are specified for each node in the corresponding table cell. Nodes for which values are left free are denoted by stars (*). Initial states can be reordered, deleted and duplicated using the buttons just above the table. Here, a unique initial state has been defined, but not selected for simulation: the state `0111` (i.e., with p53 set to 0, and the three other nodes set to 1). Note that *M1* emphasizes the fact that the value 1 is the maximal level for Mdm2nuc and for DNAdam. Several parameter configurations can be created and stored using the + button on the left side.

change the default style or define new styles. To change the graph layout, drag a node to change its position or drag an arc to create a new intermediate point. An existing intermediate point can be moved or deleted using right-click.

3.2.6. Node Ordering

Selecting the Modeling Attributes tab, with no object selected in the main window, verify that the order of the nodes is: p53, Mdm2cyt, Mdm2nuc, DNAdam. If this is not the case, modify the node order accordingly, using the arrows close to the node list at the left of the *Modelling attribute* tab. Using this node order will ease the comparison of your results with the Figures hereafter.

3.2.7. Save Your Model!

The model along with simulations settings (see hereafter) can be saved into a compressed archive (with a zginml extension) by using the *Save* option in the *File* menu. Save the model regularly during its encoding, as there is no undo functionality.

3.3. Dynamical Analysis

The qualitative state of a logical model is defined by the activity levels of its nodes. At a given state, the rules associated with each node define its *target level*. When the current level of a node is different from its target level, it is called to update toward this target level, resulting in a transition to another state. Several nodes can be called for update at a given state.

Two main strategies are then commonly used. Under the *synchronous updating*, all concerned nodes change their levels simultaneously in a unique transition toward a single successor state. In contrast, the *asynchronous updating* generates a successor state for each single node update. If the current state involves k updating calls, it will thus have k successors, each differing from the current state by the level of a single node (see Note 2 for additional explanations). The introduction of priority classes allows to define subtler updating schedules (see Note 3 and Fauré et al., 2006).

The resulting state transitions define another type of graph called *state transition graph* (STG), which represents the dynamical behavior of the logical model (i.e., the regulatory graph + logical rules). In this graph, the nodes correspond to logical states, while the arcs represent state transitions induced by the rules along with the updating scheme. Using the default level layout of GINsim for state transition graphs, it is easy to spot the stable states, defined as nodes with no outgoing arcs, displayed at the bottom. More complex attractors, defined as terminal *strongly connected components* (SCCs, maximal sets of nodes that are mutually reachable) denote oscillatory behaviors, which are harder to grasp visually.

Beyond the identification of attractors, we are particularly interested in knowing which of them can be reached from specific initial conditions. Such questions can be addressed by verifying the existence of *trajectories* (i.e., sequences of transitions), e.g., from initial states to attractor states.

3.3.1. Configuring a Simulation

Selecting the *Run Simulation* option in the *Tools* menu opens a panel enabling the construction of the dynamics (see **Figure 5**).

The boxes on the top of the panel labeled by *Select a perturbation* and *Select a reduction* permit to define (by clicking on the *Configure* buttons) and select (using the scrolling menus) model perturbations and reductions (see below).

The bottom left panel enables the definition and the recording of different parameter settings, which greatly facilitates the reproduction of simulation results. One can create, delete and reorder parameter settings by using the buttons on the right of the panel listing the parameter settings.

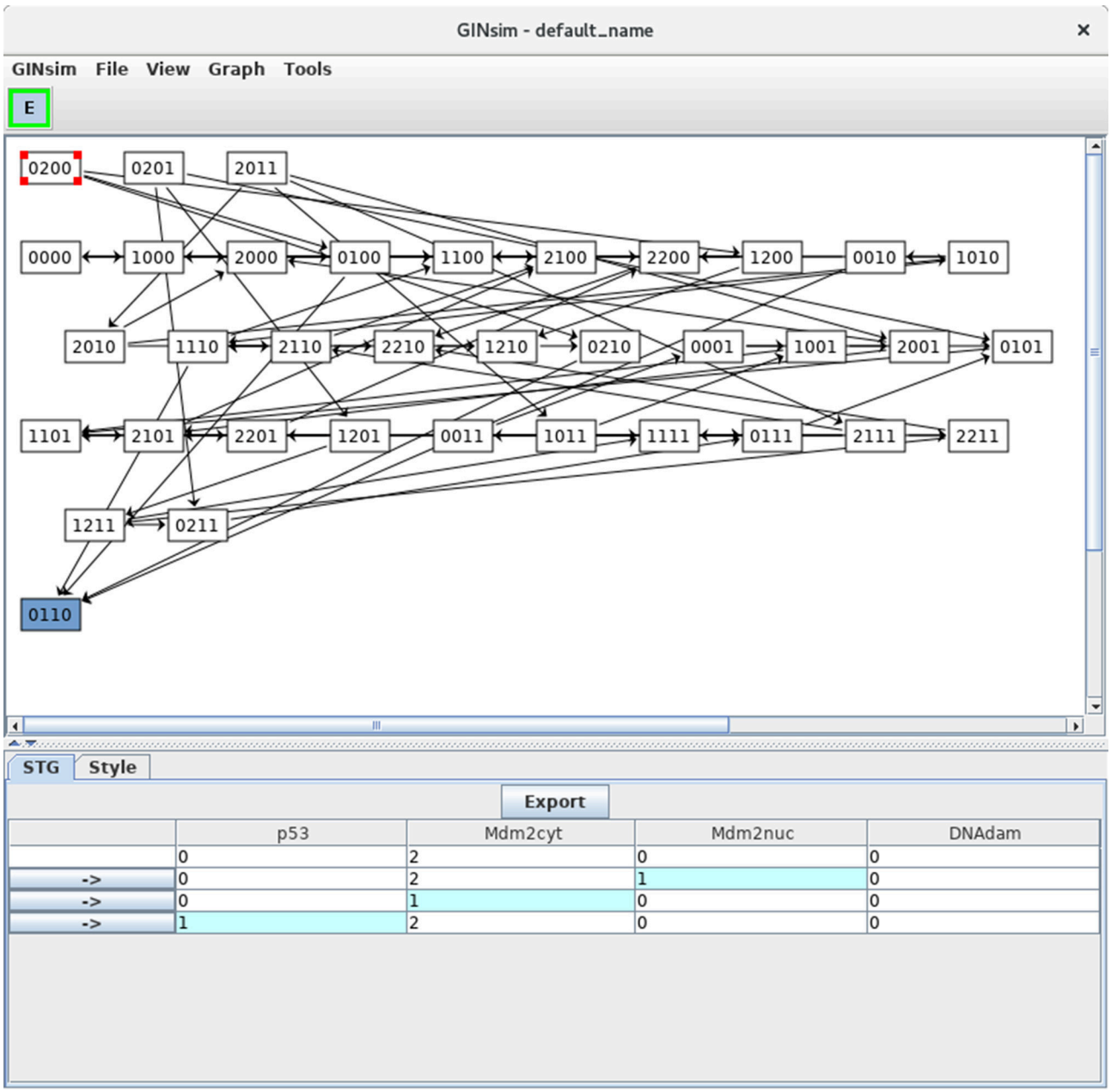


FIGURE 6 | Asynchronous state transition graph for the p53-Mdm2 model. This STG has been generated with the simulation parameters shown in **Figure 5**. The unique stable state `0110` lays at the bottom. The selected state `0200` is shown in the STG tab, with its successors.

Regarding the construction strategy, a scrolling menu enables the choice between the generation of a *state transition graph* (STG), its compression into a *strongly connected components graph* (SCC), or its further compression into a *hierarchical transition graph* (HTG) (for more details about these STG compressions, see Bérenguier et al., 2013). Using another scrolling menu, the user can select the synchronous or asynchronous updating, or define or select predefined priority classes (see Note 3 for more details on priority classes).

Finally, the *Initial State* box enables the definition and/or the selection of initial state(s), from which the construction of the dynamics will be performed. Initial states can be combined with defined sets of *Fixed inputs* (defined in the panel just below). If no initial state is selected or specified, all the states will be considered in the simulation, leading to the construction of a full STG. As the number of possible states doubles with each additional (Boolean) node, the computation of the full STG is discouraged for models involving more than 15 nodes.

3.3.2. Asynchronous Simulations

Let us first consider the construction of the *asynchronous dynamics*. Before launching the simulation, check that the default settings are specified as in **Figure 5**: state transition graph, asynchronous updating, no perturbation selected, no initial state selected. To ease comparisons with the figures enclosed in this protocol, verify that the order of the nodes is: p53, Mdm2cyt, Mdm2nuc, DNAdam in any panel listing the four components. If the order is different, it can be modified by using the green arrows displayed on the right of the list of nodes in *Modeling Attributes* panel, when no component or arc is selected.

Clicking on the *Run* button launches the simulation, i.e., the computation of the state transition graph (STG). A dialog indicates that the result is available, allowing to display the STG or to perform other actions on it. In the default level layout, the nodes with no incoming arc are placed at the top, whereas the nodes with no outgoing arc (i.e., stable states) are placed at the bottom. Stable states are further emphasized with a specific graphical attribute. In this new window, nodes can be rearranged, either manually or by selecting a predefined layout in the *View* menu. Outgoing transitions are displayed when selecting a state, as shown in **Figure 6**. Graphical settings can be modified after selecting the *Style* tab. Note that the scrolling menus propose various options, including path search functions, etc.

In **Figure 6**, the state `0200` (i.e., with high level of Mdm2cyt, and the other three nodes OFF) is selected, from which three unitary transitions are enabled by the logical rules (**Table 3**): increase of Mdm2nuc from 0 to 1, decrease of Mdm2cyt from 2 to 1, and increase of p53 from 0 to 1. The selected state and its three successor states are shown in the bottom panel. It is possible to follow a transition path by clicking on a rightwards arrow button in the bottom panel, which switches the selection to the corresponding state. When the selected state also connects to predecessors states, these are also shown, preceded by leftwards arrows.

Note that a unique stable state was obtained, `0110` (following the order defined above, this vector states that p53 = 0, Mdm2cyt = 1, Mdm2nuc = 1 and DNAdam = 0), which corresponds to the cell rest state (no p53, medium levels of cytoplasmic and nuclear Mdm2, no DNA damage).

3.3.3. Direct Computation of Stable States

Select the *Compute stable states* option in the *Tools* menu of the main window to verify that the unique stable state of this model is indeed `0110` (see **Figure 7**).

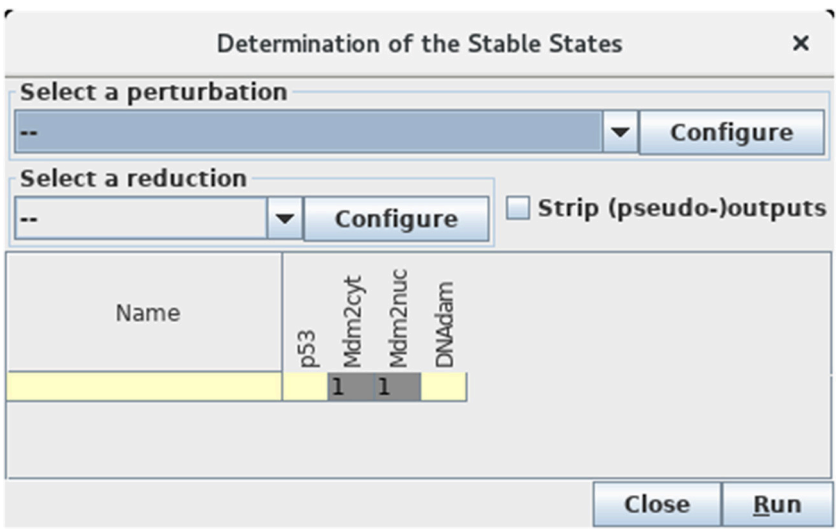


FIGURE 7 | Determination of stable states. This window appears upon selection of *Compute stable states* with the *Tools* scrolling menu. After hitting the *Run* button, GINsim returns all stable states using an efficient algorithm. In the wild type case, we obtain a unique stable state `0110` as shown (yellow and gray cells denote levels 0 and 1, respectively).

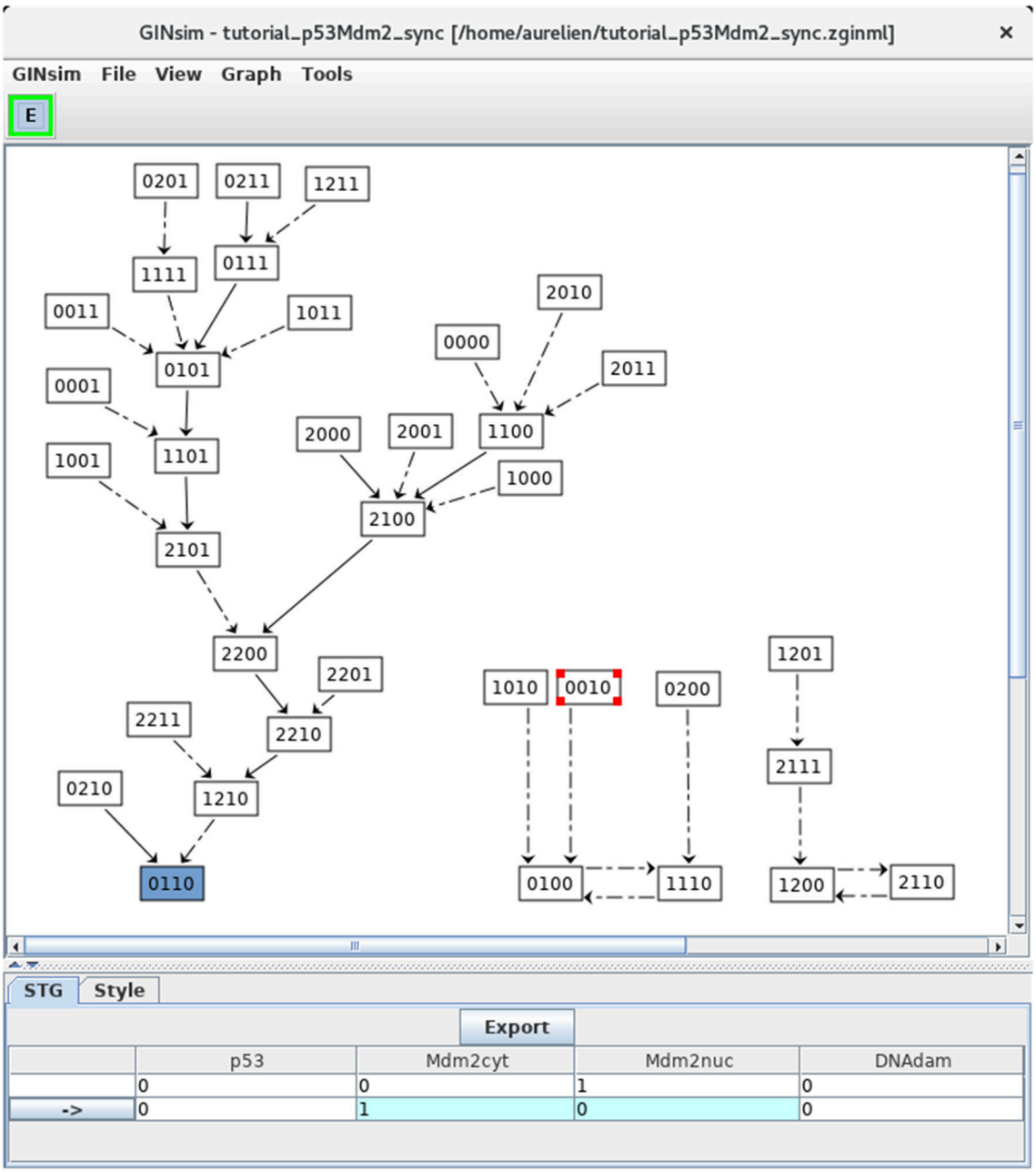


FIGURE 8 | Synchronous state transition graph for the p53-Mdm2 model. This STG has been generated with the simulation parameters shown in **Figure 5** (without specifying any initial state, but using the synchronous updating scheme). Note that the layout has been manually rearranged for sake of clarity. The STG is composed of three non connected subgraphs. On the left, we find back the resting stable state `0110`, which can be reached from 26 other states. On the right, we see that the synchronous updating further generates two two-states cyclic attractors, which can be reached from three or two other states, respectively. Solid and dotted arrows denote single and multiple transitions, respectively.

This calculation uses an algorithm bypassing the construction of the STG, which is particularly useful for large models (for more details, see Naldi et al., 2007).

If another (or no) stable state is obtained, check carefully the maximum level of each node, the threshold associated with each interaction, as well as each logical rule, as there must be a mistake somewhere...

3.3.4. Synchronous Simulations

For comparison, let us now build the state transition graph of the model using the synchronous updating strategy. Select *Run simulation* in the *Tools* menu of the main window, then select the *Synchronous* option with the scrolling menu under *Updating Mode* in **Figure 5**, and launch the simulation by clicking on the *Run* button.

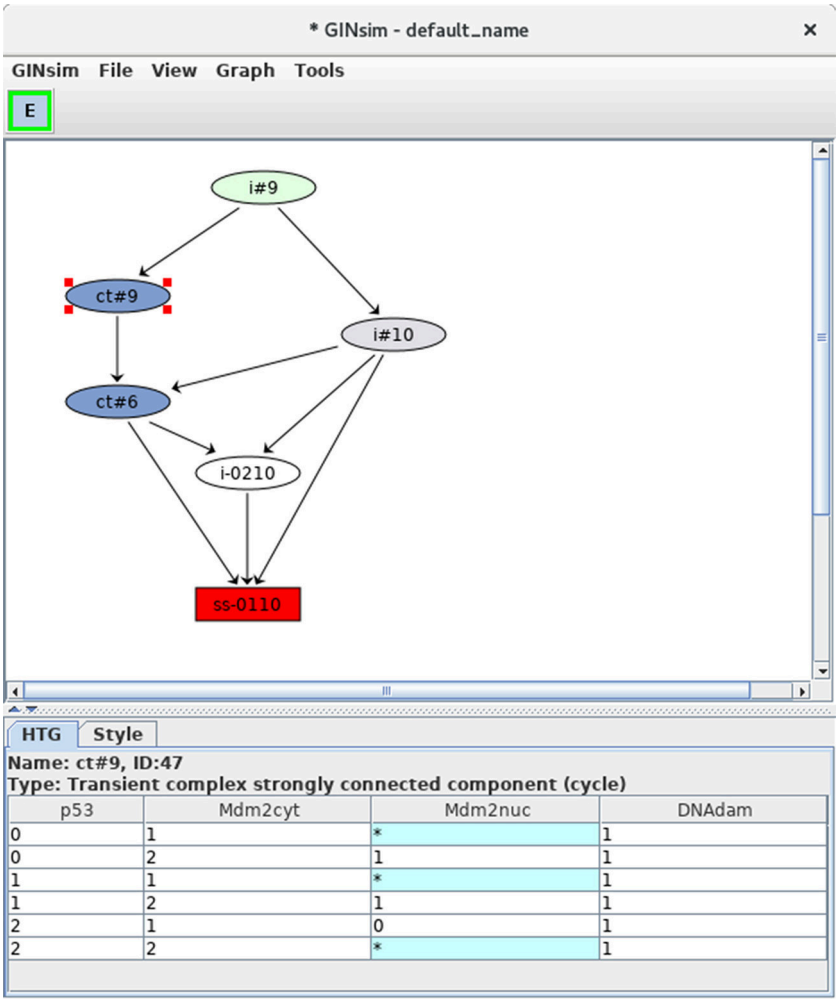


FIGURE 9 | Hierarchical transition graph. The hierarchical transition graph for the complete asynchronous dynamics of the p53-Mdm2 model is shown. It has been obtained by selecting the construction of *Hierarchical Transition Graph* in the corresponding scrolling menu when launching the simulation. Note that the layout has been manually improved. The blue nodes correspond to the two non trivial strongly connected components of the STG, and the unique stable state is shown in red at the bottom. The blue node labeled by *ct#9* has been selected; this *transient cyclic component* encompasses nine states from the STG (as indicated by the *#9* in its name), which are listed in the bottom. The * denotes all possible values for the corresponding node. Hence the first row in the table listing the states encompassed by the hypernode *ct#9* corresponds to two states: `0101` and `0111`.

The resulting STG (after a manual improvement of the layout) is shown in shown in **Figure 8**. Naturally, the stable state `0110` is preserved (bottom left), but two cyclic attractors (bottom middle and right) are now obtained. Transitions representing single and multiple node updates are denoted by solid and dotted arcs, respectively.

Note that the selected state `0010` leads to the state `0100` through simultaneous changes of Mdm2cyt and Mdm2nuc, as shown in the bottom panel (blue cells).

3.3.5. Compression of the STG

When the size of the model increases, the state transition graph (STG) quickly becomes hard to visualize. To ease its analysis, a compression (or compaction) can be performed by grouping sets of states into hyper-nodes. The arcs connecting the resulting nodes then still correspond to state transitions. In particular, by lumping states that belong to the same *strongly connected component* (SCC, in the graph-theoretical sense), an acyclic graph is obtained. Interestingly, the resulting SCC graph preserves the reachability properties of the original graph. However, in many situations, the SCC graph results only in a moderate STG compression.

To increase STG compression and ease the interpretation of the dynamics, we have recently introduced another acyclic graph, called *hierarchical transition graph*, which further merges linear chains of states (in addition to cycles) into single nodes (Bérenguier et al., 2013). The resulting graph preserves the attractors and other important dynamical properties, but does not fully conserve reachability properties.

Selecting the corresponding option with the *Construction Strategy* scrolling menu allows to compress the dynamics by using the hierarchical transition graph (HTG) representation. **Figure 9**

shows the resulting HTG, with all other simulation parameters maintained as shown in **Figure 5**.

Although relatively modest in this case (six nodes in the HTG, to be compared with 36 nodes for the original STG), this compression can be much more impressive in cases with long alternative trajectories (see e.g., Bérenguier et al., 2013; Grieco et al., 2013). However, the computation of the HTG relies on that of the STG, with the compression done progressively. Hence, HTG computation may become intractable for large networks.

At the bottom of the HTG shown in **Figure 9**, note again the stable state `0110` (red box). In addition, two blue nodes representing strongly connected components can now be clearly seen, each labeled by *ct*, for *cyclic transient*, as both nodes are the sources of outgoing transitions.

The first of these cyclic components (ct#9) is selected and the corresponding states are listed in the bottom panel (where a star stands for all possible values for the corresponding node, which compresses the list of states). This cyclic component contains nine states, all with the DNAdam node set to 1, p53 oscillating between the values 0 and 2, Mdm2cyt oscillating between 1 and 2, and Mdm2nuc oscillating between 0 and 1. Hence, this cyclic component captures large oscillations of p53 in the presence of DNA damage.

The second cyclic component (ct#6) contains six states, with DNAdam now set to 0, with p53 and Mdm2cyt both oscillating between the values 1 and 2, and Mdm2nuc oscillating between the values 0 and 1. Hence, this cyclic component captures smaller transient p53 oscillations observed just after DNA repair.

In brief, starting from initial conditions with DNAdam = 1, the system first goes through an unspecified number of large p53 activity oscillations, followed by DNA repair (DNAdam taking the value 0) along with transient smaller p53 oscillations, and finally the return to the rest state `0110`.

3.4. Additional Analyses

Several complementary analyses can be perfomed with GINsim. Hereafter, we illustrate three main functionalities: the encoding of perturbations, an algorithm enabling the analysis of the roles of regulatory circuits, along with a model reduction tool. Further information regarding GINsim functionalities can be found in the user manual and documentation available online.

3.4.1. Definition of Perturbations

Common perturbations are easily specified within the logical framework:

- A gene knock-down is specified by driving and constraining the level of the corresponding regulatory node to the value 0.
- Ectopic expression is specified by driving and constraining the level of the corresponding node to its highest value (or possibly to a range of values greater than zero, in the case of a multi-valued node).
- Multiple perturbations can be defined by combining several such constraints.
- More subtle perturbations can be defined by more sophisticated rewriting of node rules (i.e., to change the effect of a given regulatory arc).

Various perturbations can thus be defined to account for experimental observations or to generate predictions regarding the dynamical role of specific regulatory factors or interactions.

Define a mutant corresponding to an ectopic expression of DNAdam (see **Figure 10**). Such a perturbation can be encoded

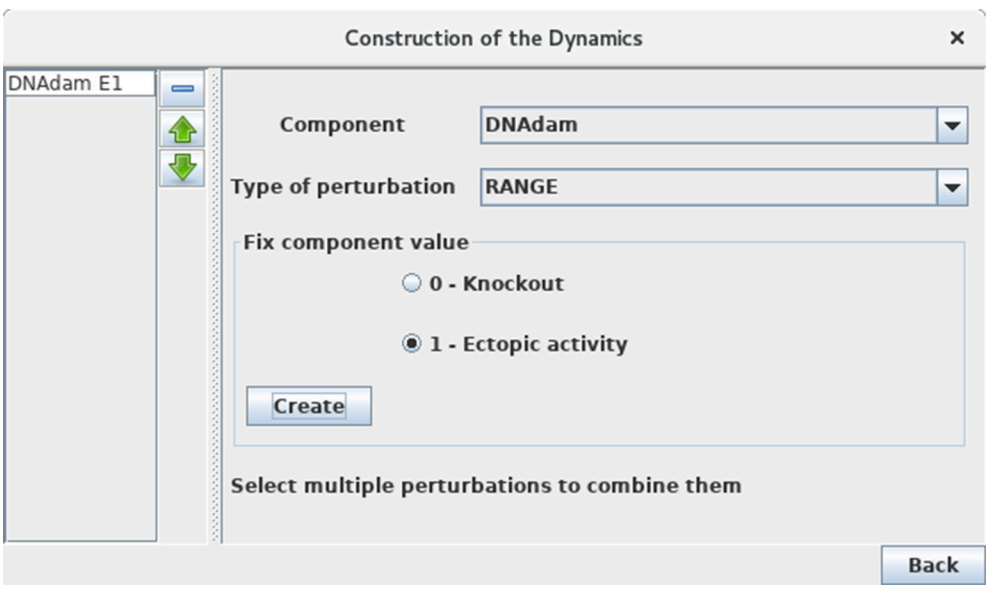


FIGURE 10 | Perturbation specification. This window can be activated from the simulation launching window (**Figure 5**) and various other windows, including the *Compute stable states* window. It enables the specification of various kinds of model perturbations, including loss-of-function and gain-of-function mutants. The figure illustrates the specification of a simple blockade of the level of DNAdam to level 1.

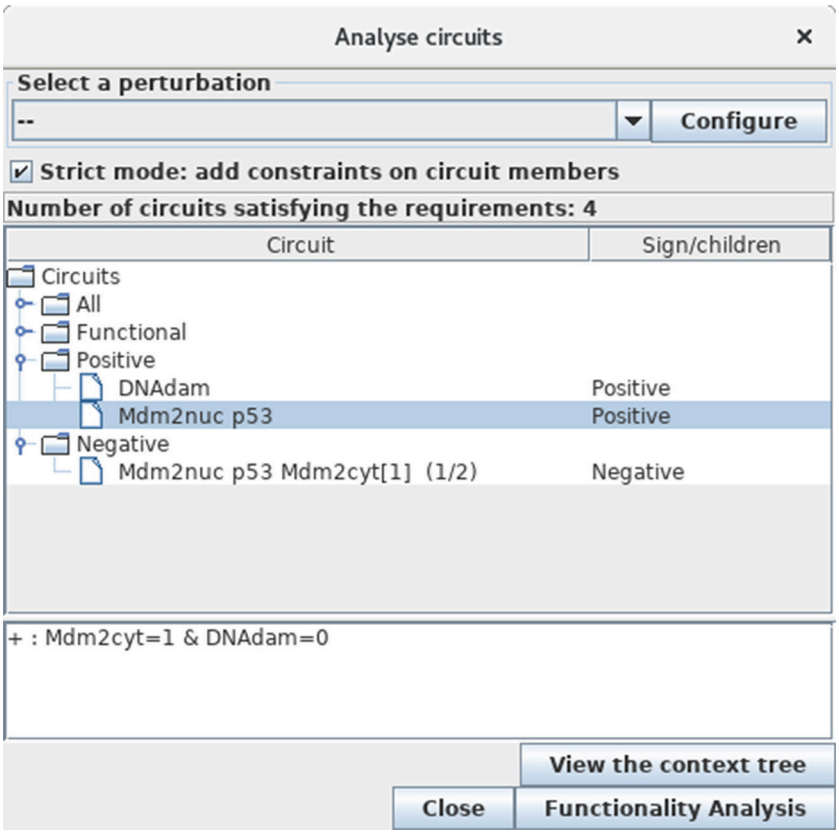


FIGURE 11 | Circuit analysis for the p53-Mdm2 logical model. This window appears after first selecting the *Analyse Circuits* option of the *Tools* scrolling menu in the main window, then clicking on the *Search Circuits* button, and finally launching the *Functionality Analysis* option. Among the four circuits found in the regulatory graph, three are functional: one is negative, while the other two are positive. The selected circuit (involving p53 and Mdm2nuc) is positive and functional when the level of Mdm2cyt is medium (equal to 1) in the absence of DNA damage (DNAdam = 0).

before the computation of stable states or of a state transition graph. Verify that the resting stable state `0110` is not stable anymore for this perturbation. Note the striking change of attractor for this perturbation, which now corresponds to ample oscillations of p53, along with oscillations of both nuclear and cytoplasmic Mdm2 forms in the presence of DNA damage.

3.4.2. Regulatory Circuit Analysis

Regulatory circuits are responsible for the emergence of dynamical properties, such as multistationarity or sustained oscillations (see Note 4). In this respect, GINsim implements specific algorithms to:

- Identify all the circuits of a regulatory graph (possibly considering constraints such as maximum length, consideration or exclusion of some nodes, etc.).
- Determine the functionality contexts of these circuits, using a computational method presented in Naldi et al. (2007).

To further identify and analyse the circuits of the model regulatory graph (see subsection 3.2), select the *Analyse Circuits* option of the *Tools* scrolling menu in the main window, then click on the *Search Circuits* button. Verify that the regulatory graph contains four circuits, among which three are functional (i.e., have a non-empty functionality context). For each functional circuit, one can verify its sign and functionality context (depending on the rules), by clicking on the *Functionality Analysis* button. As shown in **Figure 11**, the positive circuit defined by the cross inhibitions between p53 and Mdm2nuc is functional when Mdm2cyt = 1 and DNAdam = 0. Indeed, the inhibition of Mdm2nuc by p53 is not functional in the presence of DNAdam or of a high level of Mdm2cyt, or in the absence of Mdm2cyt.

3.4.3. Reduction of Logical Models

When models increase in size, it quickly becomes difficult to cope with the size of the corresponding STG. One solution consists in simplifying or reducing the model before simulation. In this respect, GINsim implements a method to reduce a model on the fly, i.e., just before the simulation. The modeler can specify the nodes to be reduced, and the logical rules associated with their targets are then recomputed taking into account the (indirect) effects of their regulators. This construction of reduced models preserves crucial dynamical properties of the original model,

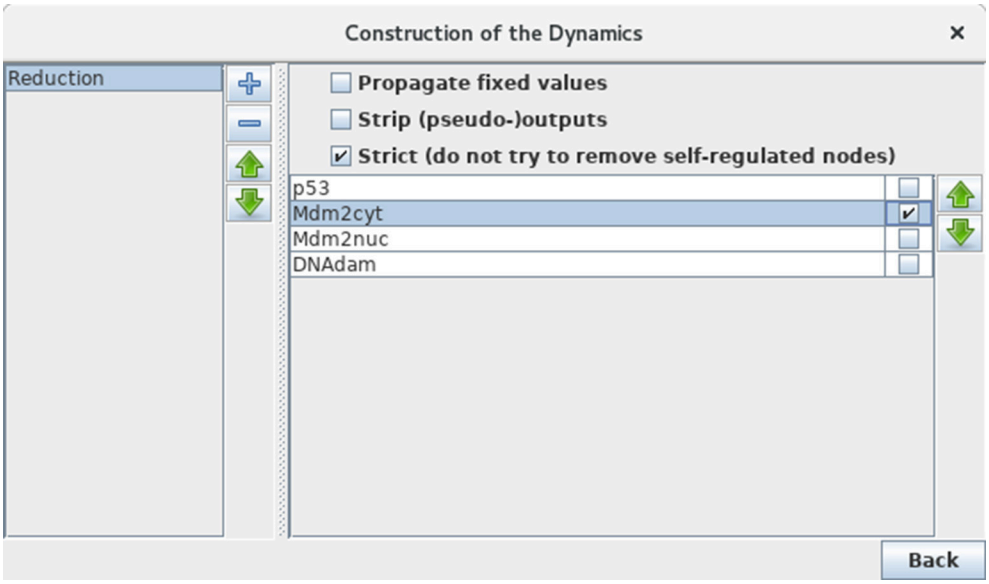


FIGURE 12 | Model reduction. This window appears following the selection of *Reduce model* from the *Tools* scrolling menu in the main GINsim window. Here, only Mdm2cyt has been selected for reduction. By hitting the *Run* Button, a reduced model is generated, provided that no self-regulated node is affected. Alternatively, one can close the window after the definition of one or several reduction(s) (the + button on the left enable to create new reductions) and then select a predefined reduction directly when performing simulations or other kinds of analyses.

including stable states and more complex attractors (Naldi et al., 2011).

Although our application is of limited size, we can still illustrate the use of GINsim model reduction functionality. Selecting the *Reduce Model* option in the *Tools* scrolling menu launches the reduction interface. Click on the + icon to define a reduction, then select the node Mdm2cyt for reduction, as shown in **Figure 12**. Clicking on the *Run* button generates a logical model encompassing only the three remaining nodes, where Mdm2nuc is the target of a dual interaction from p53. The logical rule associated with Mdm2nuc is consistently modified to take into account the former indirect effect of p53 through Mdm2cyt.

Now that a reduction has been defined, it can be selected when launching a simulation or computing stable states, without generating the reduced graph. Perform a complete asynchronous simulation to get the full state transition graph and verify that the number of states is now lower by a factor of three (12 states instead of 36) compared to **Figure 6**. Compute the HTG keeping the same parameter settings (asynchronous updating and full state space as initial condition). Although very much compressed, the resulting STG still captures the two kinds of p53 transient oscillatory behavior, ample in presence of DNA damage, smaller after DNA repair.

4. TROUBLESHOOTING

The online documentation includes a troubleshooting page (see http://doc.ginsim.org) providing some solutions to common problems. The graphical interface can have some refresh issues after long or complex modeling sessions. Such issues are usually resolved after saving the model and restarting the GINsim software. For other issues, we encourage users to send a message describing their problem to the GINsim forum or directly to the GINsim team (see http://ginsim.org/contact). Because some issues are difficult to reproduce, the user should provide log traces (using the *GINsim/support/export log files* menu option), after launching GINsim from the command line to catch additional error messages.

A few hints to solve issues that may arise in the course of this tutorial are provided below.

Some nodes can be defined as *input nodes* using a check-box in the node property panel. These input nodes can have neither incoming interactions nor regulatory rules. Indeed, input nodes have an implicit rule specifying that they maintain their current activity levels (i.e., they are maintained constant). Therefore, all regulatory interactions and rules must be removed before setting a node as an input. Likewise, the input status must be removed before adding any new regulator or rule. The model p53-Mdm2 has no input: the input check-box should be unselected for all the nodes.

In case of unexpected dynamical results (e.g., stable states, trajectories, etc.), verify successively the structure of the regulatory graph, the maximal levels of the nodes, the thresholds of the regulatory interactions with multi-valued sources and finally the regulatory rules. GINsim further provides a tool to *Compute interaction functionality*, which facilitates the identification of inconsistencies between the structure of the regulatory graph and the regulatory rules (see Note 5). To delete an invalid logical formula, select it (without editing it) and use the delete key or the contextual menu.

5. CONCLUSIONS

The logical formalism is particularly useful to model regulatory networks for which precise quantitative information is barely available, or yet to have a first glance of the dynamical properties of a complex model.

For this protocol, we have considered a network comprising four regulatory factors, and we have followed the different steps enabling the delineation of a consistent logical model. Despite its limited size, this model yields relatively complex dynamics, including several transient oscillatory patterns and a stable state. It further served as a reference to illustrate advanced functions, such as model reduction or regulatory circuit analysis.

Large signaling networks have been handled with GINsim (e.g., Calzone et al., 2010; Naldi et al., 2010; Abou-Jaoudé et al., 2015), in which input nodes denote external signals, which are not regulated and often maintained constant. Such *Input* nodes can be specified as such in GINsim to enforce the maintenance of the levels specified at initial states. As the reduction of input and output nodes or cascades have a marginal impact on the dynamics (Abou-Jaoudé et al., 2016), such reductions are facilitated in GINsim.

Furthermore, a novel functionality *Assess Attractor Reachability* in the Tool menu enables to evaluate the reachability of attractors based on stochastic simulation algorithms (for more details, see Mendes et al., 2014).

Taking advantage of the multiple export formats supported by GINsim, it is also possible to use complementary tools, including stochastic simulation software (e.g., MaBoSS, see Stoll et al., 2017), model checking tools (e.g., NuSMV, see Abou-Jaoudé et al., 2015; Abou-Jaoudé et al., 2016; Traynard et al., 2016), or yet various graph visualization and analysis packages (see Note 6 for a list of export options).

As mentioned in the introduction, various logical models for different cellular processes have been proposed during the last decades, many of them available in the repository included along with GINsim on the dedicated website (http://ginsim.org). The interested reader can thus download the model of his choice and play with it, reproduce some of the results reported in the corresponding publication, or modify and extend it according to his own research aims.

6. NOTES

1. Logical parameters constitute an alternative way of defining regulatory rules. For each node, each combination of incoming interactions then defines a logical parameter. This includes the situation in the absence of any specific activation or inhibition, or *basal level*. As a large fraction of the parameters are usually set to zero, this is the default value in GINsim (i.e., any parameter lacking an explicitly assigned value is set to 0). Consult the online documentation for details on parameters definition (http://doc.ginsim.org/lrg-parameters.html).
2. Transitions between states of the state transition graphs amount to the update of one (in the asynchronous case) or several (in the synchronous case) regulatory nodes. GINsim further support a *complete* updating mode, considering all possible (single or multiple) transitions enabled by the rules, as well as a *sequential* updating mode, which updates nodes sequentially following the predefined order node. In any case, the update (increase or decrease) of a node is unitary (current value $+1$ or -1). Obviously, this remark applies only for multi-valued nodes (for which the maximal level is greater than 1).
3. Priority classes allow to refine the updating schemes applied to construct the state transition graphs (Fauré et al., 2006). GINsim users can group nodes into different classes and assign a priority rank to each of them. In case of concurrent updating transitions (i.e., calls for level changes for several regulatory nodes in the same state), GINsim updates the node(s) belonging to the class with the highest ranking. For each priority class, the user can further specify the desired updating assumption, which then determines the treatment of concurrent transition calls inside that class. When several classes have the same rank, concurrent transitions are treated under an asynchronous assumption (no priority).
4. A regulatory circuit is defined as a sequence of interactions forming a simple closed directed path. The sign of a circuit is given by the product of the signs of its interactions. Consequently, a circuit is positive if it has an even number of inhibitions, it is negative otherwise. R. Thomas proposed that positive circuits are necessary to generate multistationarity, whereas negative circuits are necessary to generate stable oscillations (see Thieffry, 2007 and references therein). External regulators might prevent the functioning of a circuit imbedded in a more complex network. Naldi et al. (2007) proposed a method to determine the *functionality context* of a circuit in terms of constraints on the levels of its external regulator. A circuit functionality context can be interpreted as the part of the state space where the circuit is functional, i.e., generates the expected dynamical property (Comet et al., 2013).
5. The *Compute interaction functionality* option of the *Tools* scrolling menu allows to check if the signs of the interactions (graphically defined) comply with the regulatory rules. Inconsistencies arise when, for instance, a positive interaction has been drawn, while the regulatory rule of the target node defines an inhibitory effect or no effect at all. This is a convenient tool to check model inconsistencies. Note however that such inconsistencies do not prevent (inconsistent) model simulation or analysis.
6. GINsim allows the user to export logical regulatory graphs (or state transition graphs) toward various formats, facilitating the use of other software:
   - SBML-qual, the qualitative extension of the popular model exchange format (Chaouiya et al., 2013).
   - MaBoSS, a C++ software for simulating continuous/discrete time Markov processes, applied on a Boolean networks (https://maboss.curie.fr/).
   - BoolSim (http://www.vital-it.ch/software/genYsis/).

- GNA, a software for the piecewise linear modeling of regulatory networks (http://ibis.inrialpes.fr/article122.html).
- NuSMV, a symbolic model-checking tool (http://nusmv.fbk.eu/).
- Integrated Net Analyzer (INA) supporting the analysis of Place/Transition Nets (Petri Nets) and Colored Petri nets (http://www2.informatik.hu-berlin.de/~starke/ina.html).
- Snoopy, a tool to design and animate hierarchical graphs, among others Petri nets (http://www-dssz.informatik.tu-cottbus.de/DSSZ/Software/Snoopy).
- Graphviz, an open source graph visualization software offering main graph layout programs (http://www.graphviz.org/).
- Cytoscape, a popular open source software platform for visualizing molecular interaction networks (http://www.cytoscape.org/).
- Scalable Vector Graphics (SVG) format, an XML standard for describing two-dimensional graphics (http://www.w3.org/Graphics/SVG/).

AUTHOR CONTRIBUTIONS

While AN and PM have been the main developers of GINsim over the last years, all authors of this manuscript have taken part in various practical tutorials introducing the usage of GINsim to biologists, which served as a basis for the preparation of this method article. All authors have further participated in the writing of the manuscript and in the preparation of the Figures. AN, CH, WA-J, and PM should be considered as co-first authors, and CC and DT as co-last authors. All authors reviewed the content of this article and agreed to endorse it.

FUNDING

CC and PM acknowledge support from the Fundação para a Ciência e a Tecnologia, through grants PTDC/BEX-BCB/0772/2014 and PTDC/EEI-CTP/2914/2014. DT acknowledges support from the French Plan Cancer, in the context of the projects CoMET (2014–2017) and SYSTAIM (2015–2019), as well as from the French Agence Nationale pour la Recherche, in the context of the project SCAPIN [ANR-15-CE15-0006-01].

ACKNOWLEDGMENTS

The authors acknowledge numerous constructive comments from GINsim users over the years, in particular insightful feedback from Laurence Calzone, Samuel Collombet, Karla Corral, Adrien Fauré, Swann Floc'hlay, Asmund Floback, Anna Niarakis, Elisabeth Remy, Otoniel Rodriguez, and Gautier Stoll.

REFERENCES

Abou-Jaoudé, W., Monteiro, P. T., Naldi, A., Grandclaudon, M., Soumelis, V., Chaouiya, C., et al. (2015). Model checking to assess T-Helper cell plasticity. *Front. Bioeng. Biotechnol.* 2:86. doi: 10.3389/fbioe.2014.00086

Abou-Jaoudé, W., Ouattara, D. A., and Kaufman, M. (2009). From structure to dynamics: frequency tuning in the p53-Mdm2 network I. logical approach. *J. Theor. Biol.* 258, 561–577. doi: 10.1016/j.jtbi.2009.02.005

Abou-Jaoudé, W., Traynard, P., Monteiro, P. T., Saez-Rodriguez, J., Helikar, T., Thieffry, D., et al. (2016). Logical modeling and dynamical analysis of cellular networks. *Front. Genet.* 7:94. doi: 10.3389/fgene.2016.00094

Azpeitia, E., Davila-Velderrain, J., Villarreal, C., and Alvarez-Buylla, E. R. (2014). "Gene regulatory network models for floral organ determination," in *Flower Development,* eds J. L. Riechmann and F. Wellmer (New York, NY: Humana Press), 441–469.

Barak, Y., Juven, T., Haffner, R., and Oren, M. (1993). mdm2 expression is induced by wild type p53 activity. *EMBO J.* 12, 461–468.

Bérenguier, D., Chaouiya, C., Monteiro, P. T., Naldi, A., Remy, E., Thieffry, D., (2013). Dynamical modeling and analysis of large cellular regulatory networks. *Chaos* 23:025114. doi: 10.1063/1.4809783

Brooks, C. L., and Gu, W. (2006). p53 ubiquitination: Mdm2 and beyond. *Mol. Cell* 21, 307–315. doi: 10.1016/j.molcel.2006.01.020

Calzone, L., Tournier, L., Fourquet, S., Thieffry, D., Zhivotovsky, B., Barillot, E., et al. (2010). Mathematical modelling of cell-fate decision in response to death receptor engagement. *PLoS Comput. Biol.* 6:e1000702. doi: 10.1371/journal.pcbi.1000702

Chaouiya, C., Bérenguier, D., Keating, S. M., Naldi, A., van Iersel, M. P., Rodriguez, N., et al. (2013). SBML qualitative models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools. *BMC Syst. Biol.* 7:135. doi: 10.1186/1752-0509-7-135

Choi, M., Shi, J., Jung, S. H., Chen, X., and Cho, K. H. (2012). Attractor landscape analysis reveals feedback loops in the p53 network that control the cellular response to DNA damage. *Sci. Signal.* 5:ra83. doi: 10.1126/scisignal.2003363

Ciliberto, A., Novak, B., and Tyson, J. J. (2005). Steady states and oscillations in the p53/Mdm2 network. *Cell Cycle* 4, 488–493. doi: 10.4161/cc.4.3.1548

Collombet, S., van Oevelen, C., Sardina Ortega, J. L., Abou-Jaoudé, W., Di Stefano, B., Thomas-Chollier, M., et al. (2017). Logical modeling of lymphoid and myeloid cell specification and transdifferentiation. *Proc. Natl. Acad. Sci. U.S.A.* 114, 5792–5799. doi: 10.1073/pnas.1610622114

Comet, J.-P., Noual, M., Richard, A., Aracena, J., Calzone, L., Demongeot, J., et al. (2013). On circuit functionality in boolean networks. *Bull. Math. Biol.* 75, 906–919. doi: 10.1007/s11538-013-9829-2

Coolen, M., Thieffry, D., Drivenes, Ø., Becker, T. S., and Bally-Cuif, L. (2012). miR-9 controls the timing of neurogenesis through the direct inhibition of antagonistic factors. *Dev. Cell* 22, 1052–1064. doi: 10.1016/j.devcel.2012.03.003

Coutts, A. S., Boulahbel, H., Graham, A., and La Thangue, N. B. (2007). Mdm2 targets the p53 transcription cofactor JMY for degradation. *EMBO Reports* 8, 84–90. doi: 10.1038/sj.embor.7400855

Fauré, A., Naldi, A., Chaouiya, C., and Thieffry, D. (2006). Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle. *Bioinformatics* 22, 124–31. doi: 10.1093/bioinformatics/btl210

Fauré, A., Naldi, A., Lopez, F., Chaouiya, C., Ciliberto, A., and Thieffry, D. (2009). Modular logical modelling of the budding yeast cell cycle. *Mol. Biosyst.* 5, 1787–1796. doi: 10.1039/b910101m

Fauré, A., Vreede, B. M., Sucena, E., and Chaouiya, C. (2014). A discrete model of drosophila eggshell patterning reveals cell-autonomous and juxtacrine effects. *PLoS Comput. Biol.* 10:e1003527. doi: 10.1371/journal.pcbi.1003527

Flobak, Å., Baudot, A., Remy, E., Thommesen, L., Thieffry, D., Kuiper, M., et al. (2015). Discovery of drug synergies in gastric cancer cells predicted by logical modeling. *PLoS Comput. Biol.* 11:e1004426. doi: 10.1371/journal.pcbi.1004426

Gatz, S. A., and Wiesmüller, L. (2006). p53 in recombination and repair. *Cell Death Differ.* 13, 1003–1016. doi: 10.1038/sj.cdd.4401903

González, A., Chaouiya, C., and Thieffry, D. (2006). Dynamical analysis of the regulatory network defining the dorsal-ventral boundary of the drosophila wing imaginal disc. *Genetics* 174, 1625–1634. doi: 10.1534/genetics.106.061218

González, A., Chaouiya, C., and Thieffry, D. (2008). Logical modelling of the role of the Hh pathway in the patterning of the Drosophila wing disc. *Bioinformatics* 24, i234–i240. doi: 10.1093/bioinformatics/btn266

Grieco, L., Calzone, L., Bernard-Pierrot, I., Radvanyi, F., Kahn-Perlès, B., and Thieffry, D. (2013). Integrative modelling of the influence of MAPK network on cancer cell fate decision. *PLoS Comput. Biol.* 9:21003286. doi: 10.1371/annotation/90e5e4be-952b-42b8-b56d-46baae3479ed

Iwamoto, K., Hamada, H., Eguchi, Y., and Okamoto, M. (2014). Stochasticity of intranuclear biochemical reaction processes controls the final decision of cell fate associated with DNA damage. *PLoS ONE* 9:e101333. doi: 10.1371/journal.pone.0101333

Le Novère, N., Hucka, M., Mi, H., Moodie, S., Schreiber, F., Sorokin, A., et al. (2009). The systems biology graphical notation. *Nat. Biotechnol.* 27, 735–741. doi: 10.1038/nbt.1558

Martinez-Sanchez, M. E., Mendoza, L., Villarreal, C., and Alvarez-Buylla, E. R. (2015). A minimal regulatory network of extrinsic and intrinsic factors recovers observed patterns of CD4+ T cell differentiation and plasticity. *PLoS Comput. Biol.* 11:e1004324. doi: 10.1371/journal.pcbi.1004324

Mayo, L. D. and Donner, D. B. (2002). The PTEN, Mdm2, p53 tumor suppressor-oncoprotein network. *Trends Biochem. Sci.* 27, 462–467. doi: 10.1016/S0968-0004(02)02166-7

Mbodj, A., Gustafson, E. H., Ciglar, L., Junion, G., Gonzalez, A., Girardot, C., et al. (2016). Qualitative dynamical modelling can formally explain mesoderm specification and predict novel developmental phenotypes. *PLOS Comput. Biol.*, 12:e1005073. doi: 10.1371/journal.pcbi.1005073

Mendoza, L. and Méndez, A. (2015). A dynamical model of the regulatory network controlling lymphopoiesis. *Biosystems* 137, 26–33. doi: 10.1016/j.biosystems.2015.09.004

Mendes, N. D., Monteiro, P. T., Carneiro, J., Remy, E., and Chaouiya, C. (2014). Quantification of reachable attractors in asynchronous discrete dynamics. *arXiv*:1411.3539

Mendoza, L., Thieffry, D., and Alvarez-Buylla, E. R. (1999). Genetic control of flower morphogenesis in Arabidopsis thaliana: a logical analysis. *Bioinformatics* 15, 593–606. doi: 10.1093/bioinformatics/15.7.593

Naldi, A., Carneiro, J., Chaouiya, C., and Thieffry, D. (2010). Diversity and plasticity of th cell types predicted from regulatory network modelling. *PLoS Comput. Biol.* 6:e1000912. doi: 10.1371/journal.pcbi.1000912

Naldi, A., Monteiro, P., Müssel, C., Kestler, H. A., Thieffry, D., Xenarios, I., et al. (2015). Cooperative development of logical modelling standards and tools with CoLoMoTo. *Bioinformatics* 31, 1154–1159. doi: 10.1093/bioinformatics/btv013

Naldi, A., Remy, E., Thieffry, D., and Chaouiya, C. (2011). Dynamically consistent reduction of logical regulatory graphs. *Theor. Comput. Sci.* 412, 2207–2218. doi: 10.1016/j.tcs.2010.10.021

Naldi, A., Thieffry, D., and Chaouiya, C. (2007). Decision diagrams for the representation of logical models of regulatory networks. *Lect. Notes Comput. Sci.* 4695, 233–247. doi: 10.1007/978-3-540-75140-3_16

Oliner, J. D., Pietenpol, J. A., Thiagalingam, S., Gyuris, J., Kinzler, K. W., and Vogelstein, B. (1993). Oncoprotein MDM2 conceals the activation domain of tumour suppressor p53. *Nature* 362, 857–860. doi: 10.1038/362857a0

Ouattara, D. A., Abou-Jaoude, W., and Kaufman, M. (2010). From structure to dynamics: frequency tuning in the p53-Mdm2 network. II Differential and stochastic approaches. *J. Theor. Biol.* 264, 1177–1189. doi: 10.1016/j.jtbi.2010.03.031

Puszynski, K., Hat, B., and Lipniacki, T. (2008). Oscillations and bistability in the stochastic model of p53 regulation. *J. Theor. Biol.* 254, 452–465. doi: 10.1016/j.jtbi.2008.05.039

Remy, E., Rebouissou, S., Chaouiya, C., Zinovyev, A., Radvanyi, F., and Calzone, L. (2015). A modeling approach to explain mutually exclusive and co-occurring genetic alterations in ladder tumorigenesis. *Cancer Res.* 75, 4042–4052. doi: 10.1158/0008-5472.CAN-15-0602

Sahin, O., Fröhlich, H., Löbke, C., Korf, U., Burmester, S., Majety, M., et al. (2009). Modeling ERBB receptor-regulated G1/S transition to find targets for de novo trastuzumab resistance. *BMC Syst. Biol.* 3:1. doi: 10.1186/1752-0509-3-1

Sánchez, L., Chaouiya, C., and Thieffry, D. (2008). Segmenting the fly embryo: a logical analysis of the segment polarity cross-regulatory module. *Int. J. Dev. Biol.* 52, 1059–1075. doi: 10.1387/ijdb.072439ls

Sánchez, L., and Thieffry, D. (2001). A logical analysis of the drosophila gap-gene system. *J. Theor. Biol.* 211, 115–141. doi: 10.1006/jtbi.2001.2335

Sánchez, L., and Thieffry, D. (2003). Segmenting the fly embryo: a logical analysis of the pair-rule cross-regulatory module. *J. Theor. Biol.* 224, 517–537. doi: 10.1016/S0022-5193(03)00201-7

Stoll, G., Caron, B., Viara, E., Dugourd, A., Zinovyev, A., Naldi, A., et al. (2017). MaBoSS 2.0: an environment for stochastic Boolean modeling. *Bioinformatics* 33, 2226–2228. doi: 10.1093/bioinformatics/btx123

Sun, T., and Cui, J. (2014). A plausible model for bimodal p53 switch in DNA damage response. *FEBS Lett.* 588, 815–821. doi: 10.1016/j.febslet.2014.01.044

Thieffry, D. (2007). Dynamical roles of biological regulatory circuits. *Brief. Bioinform.* 8, 220–225. doi: 10.1093/bib/bbm028

Thieffry, D., and Thomas, R. (1995). Dynamical behaviour of biological regulatory networks, II. Immunity control in bacteriophage lambda. *Bull. Math. Biol.* 57, 277–297.

Thomas, R. (1991). Regulatory networks seen as asynchronous automata: a logical description. *J. Theor. Biol.* 153, 1–23. doi: 10.1016/S0022-5193(05)80350-9

Thomas, R., Thieffry, D., and Kaufman, M. (1995). Dynamical behaviour of biological regulatory networks i. biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bull. Math. Biol.* 57, 247–276. doi: 10.1007/BF02460618

Traynard, P., Fauré, A., Fages, F., and Thieffry, D. (2016). Logical model specification aided by model-checking techniques: application to the mammalian cell cycle regulation. *Bioinformatics* 32, i772–i780. doi: 10.1093/bioinformatics/btw457

Vogelstein, B., Lane, D., and Levine, A. J. (2000). Surfing the p53 network. *Nature* 408, 307–310. doi: 10.1038/35042675

Zhang, X. P., Liu, F., and Wang, W. (2011). Two-phase dynamics of p53 in the DNA damage response. *Proc. Natl. Acad. Sci. U.S.A.* 108, 8990–8995. doi: 10.1073/pnas.1100600108

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

METHODS
published: 19 June 2018
doi: 10.3389/fphys.2018.00680

The CoLoMoTo Interactive Notebook: Accessible and Reproducible Computational Analyses for Qualitative Biological Networks

Aurélien Naldi[1], *Céline Hernandez*[1], *Nicolas Levy*[2,3], *Gautier Stoll*[4,5,6,7,8], *Pedro T. Monteiro*[9], *Claudine Chaouiya*[10], *Tomáš Helikar*[11], *Andrei Zinovyev*[12,13,14,15], *Laurence Calzone*[12,13,14], *Sarah Cohen-Boulakia*[2], *Denis Thieffry*[1]* *and Loïc Paulevé*[2]*

[1] Computational Systems Biology Team, Institut de Biologie de l'Ecole Normale Supérieure, Centre National de la Recherche Scientifique UMR8197, Institut National de la Santé et de la Recherche Médicale U1024, École Normale Supérieure, PSL Université, Paris, France, [2] Laboratoire de Recherche en Informatique UMR8623, Université Paris-Sud, Centre National de la Recherche Scientifique, Université Paris-Saclay, Orsay, France, [3] École Normale Supérieure de Lyon, Lyon, France, [4] Université Paris Descartes/Paris V, Sorbonne Paris Cité, Paris, France, [5] Équipe 11 Labellisée Ligue Nationale Contre le Cancer, Centre de Recherche des Cordeliers, Paris, France, [6] Institut National de la Santé et de la Recherche Médicale, U1138, Paris, France, [7] Université Pierre et Marie Curie, Paris, France, [8] Metabolomics and Cell Biology Platforms, Gustave Roussy Cancer, Villejuif, France, [9] INESC-ID/Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal, [10] Instituto Gulbenkian de Ciência, Oeiras, Portugal, [11] Department of Biochemistry, University of Nebraska-Lincoln, Lincoln, NE, United States, [12] Institut Curie, PSL Research University, Paris, France, [13] Institut National de la Santé et de la Recherche Médicale, U900, Paris, France, [14] MINES ParisTech, PSL Research University, CBIO-Centre for Computational Biology, Paris, France, [15] Lobachevsky University, Nizhni Novgorod, Russia

OPEN ACCESS

Edited by:
Pierre De Meyts, de Duve Institute, Belgium

Reviewed by:
Oksana Sorokina, University of Edinburgh, United Kingdom
Kyle B. Gustafson, Naval Surface Warfare Center Carderock Division (NSWCCD), United States

***Correspondence:**
Denis Thieffry
thieffry@ens.fr
Loïc Paulevé
loic.pauleve@lri.fr

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: *05 April 2018*
Accepted: *15 May 2018*
Published: *19 June 2018*

Citation:
Naldi A, Hernandez C, Levy N, Stoll G, Monteiro PT, Chaouiya C, Helikar T, Zinovyev A, Calzone L, Cohen-Boulakia S, Thieffry D and Paulevé L (2018) The CoLoMoTo Interactive Notebook: Accessible and Reproducible Computational Analyses for Qualitative Biological Networks. Front. Physiol. 9:680. doi: 10.3389/fphys.2018.00680

Analysing models of biological networks typically relies on workflows in which different software tools with sensitive parameters are chained together, many times with additional manual steps. The accessibility and reproducibility of such workflows is challenging, as publications often overlook analysis details, and because some of these tools may be difficult to install, and/or have a steep learning curve. The CoLoMoTo Interactive Notebook provides a unified environment to edit, execute, share, and reproduce analyses of qualitative models of biological networks. This framework combines the power of different technologies to ensure repeatability and to reduce users' learning curve of these technologies. The framework is distributed as a Docker image with the tools ready to be run without any installation step besides Docker, and is available on Linux, macOS, and Microsoft Windows. The embedded computational workflows are edited with a Jupyter web interface, enabling the inclusion of textual annotations, along with the explicit code to execute, as well as the visualization of the results. The resulting notebook files can then be shared and re-executed in the same environment. To date, the CoLoMoTo Interactive Notebook provides access to the software tools GINsim, BioLQM, Pint, MaBoSS, and Cell Collective, for the modeling and analysis of Boolean and multi-valued networks. More tools will be included in the future. We developed a Python interface for each of these tools to offer a seamless integration in the Jupyter web interface and ease the chaining of complementary analyses.

Keywords: computational systems biology, reproducibility, model analysis, Boolean networks, Python programming language

1. INTRODUCTION

Recently, the scientific community has been increasingly concerned about difficulties in reproducing already published results. In the context of preclinical studies, observed difficulties to reproduce important findings have raised controversy (see e.g., Richter et al., 2010; Begley and Ellis, 2012; Smith and Houghton, 2013; Errington et al., 2014; and Begley and Ioannidis, 2015 for a review on this topic). Although not invalidating the findings, these observations have shaken the community. In 2016, a Nature survey pointed to the multi-factorial origin of this "reproducibility crisis" (Baker, 2016). Factors related to computational analyses were highlighted, in particular the unavailability of code and methods, along with the technical expertise required to reproduce the computations. The scientific community is progressively addressing this problem. Prestigious conferences (such as two major conferences from the database community, namely, VLDB[1] and SIGMOD[2]) and journals (such as PNAS, Biostatistics (Peng, 2009), Nature (Santori, 2016), and Science (Yaffe, 2015), to name only a few) now encourage or even require published results to be accompanied by all the information necessary to reproduce them.

While the reproducibility challenges have first been observed in domains where deluge of data were quickly becoming available (e.g., Next Generation Sequencing data analyses), the problem is now present in many (if not all) communities where computational analyses and simulations are performed. In particular, the Systems Biology community is facing a proliferation of approaches to perform a large variety of tasks, including the development of dynamical models, complex simulations, and multiple comparisons between varying conditions of model variants. Consequently, reproducing results from systems biology studies becomes increasingly difficult. Furthermore, although the combination of different tools would provide various new scientific opportunities, this is currently hindered by technical issues.

Several initiatives have been launched by the community to address reproducibility issues for computational modeling of biochemical networks. These include guidelines for model annotations (MIRIAM, Le Novère et al., 2005) and simulation descriptions (MIASE, Waltemath et al., 2011a), as well as standards for model exchange (SBML, Hucka et al., 2003) and simulation parametrizations (SED-ML, Waltemath et al., 2011b). This collective effort is coordinated by the *COmputational Modeling in BIology NEtwork* (COMBINE[3]).

The Consortium for Logical Models and Tools (CoLoMoTo[4]) has been organized to bring together computational modeling researchers and address the aforementioned reproducibility and reusability issues within the sub-domain of logical models and software tools (Naldi et al., 2015). As a first outcome to foster model exchange and software interoperability, the SBML L3 package qual was developed (Chaouiya et al., 2013, 2015). In this manuscript, we report the next phase of the CoLoMoTo efforts in the area of reproducibility in computational systems biology: The CoLoMoTo Interactive Notebook, which provides an easy-to-use environment to edit, execute, share, and reproduce analyses of qualitative models of biological networks by seamlessly integrating various logical modeling software tools.

The teams involved in CoLoMoTo, gathering around 50 researchers within 20 groups and laboratories, have produced various software tools for the qualitative modeling and analysis of biological networks. They are also involved in the development of novel computational methods and models. This method article presents a collective effort to provide the community with a reproducibility-oriented framework combining software tools related to logical modeling. This framework combines the power of different approaches to ensure repeatability and to reduce the requirement of technical knowledge from users. The provided Docker image facilitates the stability of a contained environment needed for repeatable computational modeling and analyses. The framework includes a set of pre-installed tools from the CoLoMoTo community. On the other hand, specific binding and interfaces integrated in a Jupyter environment reduce the learning curve and improve accessibility. The use of this framework is demonstrated by a case study in a companion protocol article, which consists in a thoroughly annotated Jupiter notebook (Levy et al., 2018)[5].

The method article is structured as follows. Section 2 provides a brief introduction to qualitative models of biological networks and to their analyses. Section 3 describes the main components (Docker image, Python programming interface, Jupyter interactive web interface) of our framework to facilitate the access to CoLoMoTo software tools, a prime prerequisite for the reproducibility of the computational analyses. Section 4 illustrates how our framework can address several challenges related to the reproducibility of computational analyses, ranging from the repeat of a sequence of analyses in the exact same software environment, to the use of alternate methods to reproduce a result. Finally, section 5 provides an introductory guide on how to use the new framework, and section 6 discusses possible extensions.

2. BACKGROUND ON QUALITATIVE DYNAMICAL MODELS AND THEIR COMPUTATIONAL ANALYSIS

Since the pioneering work of Kauffman (1969), Thomas (1973), and others, logical (e.g., Boolean) models have emerged as a framework of choice to model complex biological networks, focusing for example on the roles of transcriptional regulatory circuits in cell differentiation and development, of signaling pathways in cell fate decisions, etc. (for a review, see e.g., Abou-Jaoudé et al., 2016).

[1] International conference on Very Large Data Bases.
[2] ACM's Special Interest Group on Management Of Data.
[3] http://co.mbine.org
[4] http://colomoto.org

[5] The notebook can be previewed and downloaded at https://nbviewer.jupyter.org/github/colomoto/colomoto-docker/blob/2018-03-31/usecases/Usecase%20-%20Mutations%20enabling%20tumour%20invasion.ipynb

2.1. Qualitative Modeling

The definition of a qualitative logical model, such as a *Boolean model* usually relies first on the delineation of a *regulatory graph*, where each *node* denotes a regulatory component (e.g., a protein or a gene), while (positive or negative) *arcs* represent interactions (activation or inhibition) between their source and target nodes. Each node is modeled as a discrete variable, having a finite number of possible values, typically Boolean, i.e., only two values, 0 or 1, denoting e.g., protein absence/inactivity or presence/activity. A *Boolean function* or *rule* is then defined for each node to specify how its value may change depending on the values of its regulators.

The *state* of a network is modeled as a vector encompassing the (Boolean or multi-valued) values of all the nodes of the regulatory graph, with a prescribed ordering. The state of the network can be updated according to the logical functions defined for each node, triggering a *transition* toward a successor state. When at a given state, several nodes are called for an update, different updating modes can be considered. The *synchronous updating mode* updates all nodes simultaneously, thus leading to a unique successor state. Hence, the dynamical behavior is fully deterministic. In contrast, the *asynchronous updating mode* updates only one node, choosen non-deterministically, thus leading to different possible successor states. Several variants and extensions of these updating modes have been defined, for instance assigning pre-determined priorities or assigning probabilities to node updates, or considering simultaneous updates of sub-groups of nodes.

2.2. Dynamical Analysis

The dynamical behavior of the model can be represented as a *state transition graph*, where vertices correspond to different states of the network, and directed edges represent *transitions* between states, following a selected updating mode. *Dynamical analyses* consist then in characterizing different properties of this state transition graph.

Attractors are one of the most prominent features studied in Boolean and multi-valued networks. Attractors model the asymptotic behaviors of the system, and correspond to the *terminal strongly connected components* of the state transition graph. Attractors can be of different nature, either reduced to a single *stable state* (or *fixed point*), from which no transition is possible, or *cyclic* sequences of states, modeling sustained oscillations. From a biological point of view, computing attractors is generally particularly relevant. The presence of multiple attractors can represent alternative cell fates (such as cell differentiation states), while cyclic attractors further represent periodic behaviors (such as cell cycle or circadian rhythms). The computation of attractors is addressed by different software tools, such as BIOLQM (Naldi, in review[6]), GINSIM (Naldi et al., 2018), PINT (Paulevé, 2017), BOOLSIM (Garg et al., 2008), BOOLEANNET (Albert et al., 2008), PYBOOLNET (Klarner et al., 2017), and BOOLNET (Müssel et al., 2010).

Simulations allow capturing the states *reachable* from a given (set of) *initial state(s)*. They can consist of random walks in the complete state transition graph, take into account updating priority schemes to distinguish fast versus slow processes and thereby obtain a simpler state transition graph (Fauré et al., 2006), as implemented in the software tool BIOLQM, or rely on user-defined transition probabilities and timing, as implemented into the software tools MABOSS (Stoll et al., 2012, 2017) and CELLCOLLECTIVE (Helikar et al., 2012; Todd and Helikar, 2012).

Model checking techniques developed for software verification in computer science allow verifying formally dynamical properties on state transition graphs and are regularly employed for analysing biological systems (Batt et al., 2005; Abou-Jaoudé et al., 2015; Bartocci and Lió, 2016; Traynard et al., 2016). The properties are specified using so-called *temporal logics*, which enable the formulation of queries regarding asymptotic or transient dynamical properties, taking into account all the state transitions of the model. The accordance of a Boolean/multi-valued model with such properties is verified using a general purpose model checker such as NUSMV (Cimatti et al., 2002) to which GINSIM and PINT provide access.

It is worth noticing that the number of states of a Boolean or multi-valued network grows exponentially with the number of nodes. The above mentioned methods typically suffer from this complexity, and hence face limitations regarding network size (currently, this limit is of the order of fifty to a hundred of nodes, depending on the analysis and the complexity of the dynamics). Nevertheless, different approaches enable the analysis of large scale qualitative networks by means of structural analyses, model reductions or abstractions. The CoLoMoTo Interactive Notebook provides access to methods for *model reductions*, such as by Naldi et al. (2011), implemented in BIOLQM, which preserves stable states, while cyclic attractors and reachability can be affected in predictable ways, or by using *formal approximations* of the dynamical behavior, as implemented in PINT, which allow tackling networks with several thousands of nodes (Paulevé, 2017, in press). Other approaches include, for instance, Petri net model reduction for trajectories in signaling pathways (Talcott and Dill, 2006), subnetwork analysis (Siebert, 2009), computational algebra (Veliz-Cuba et al., 2014), and motif-based abstractions for attractors (Gan and Albert, 2018).

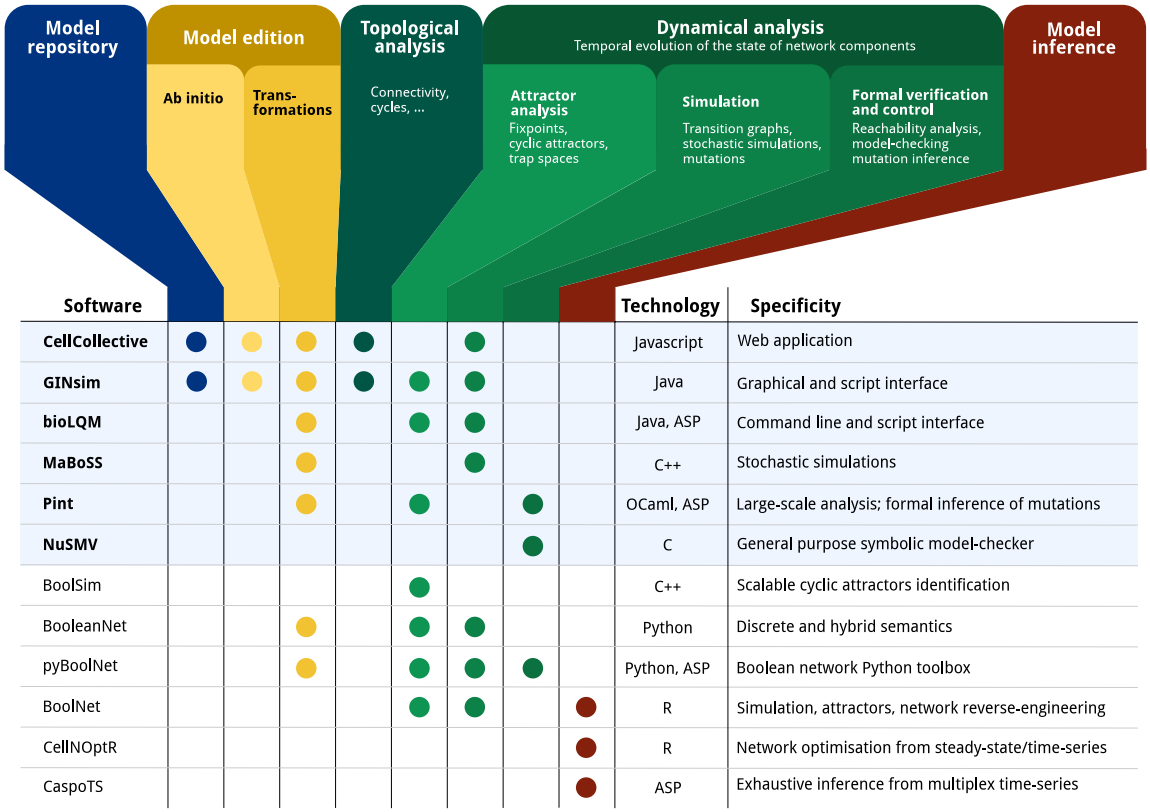
Figure 1 gives an overview of a range of software tools for the analysis of qualitative models, specifying their main features along with the main underlying technologies.

3. ACCESSIBILITY OF COLOMOTO SOFTWARE TOOLS

The CoLoMoTo Interactive Notebook aims at offering a unified environment for accessing a range of complementary software tools for the analysis of qualitative models of biological networks. To achieve such a goal, our framework relies on three complementary technologies.

First, we use the Docker system to provide images of pre-installed selected CoLoMoTo software tools, thus reducing significantly the burden of installing individually each software tool. The software installed within Docker images can be executed on GNU/Linux, macOS, and Microsoft Windows,

[6]Preprint on bioRxiv https://doi.org/10.1101/287011



Software	Technology	Specificity
CellCollective	Javascript	Web application
GINsim	Java	Graphical and script interface
bioLQM	Java, ASP	Command line and script interface
MaBoSS	C++	Stochastic simulations
Pint	OCaml, ASP	Large-scale analysis; formal inference of mutations
NuSMV	C	General purpose symbolic model-checker
BoolSim	C++	Scalable cyclic attractors identification
BooleanNet	Python	Discrete and hybrid semantics
pyBoolNet	Python, ASP	Boolean network Python toolbox
BoolNet	R	Simulation, attractors, network reverse-engineering
CellNOptR	R	Network optimisation from steady-state/time-series
CaspoTS	ASP	Exhaustive inference from multiplex time-series

FIGURE 1 | Feature matrix and characteristics of a range of software tools devoted to the qualitative modeling and analysis of biological networks: CELLCOLLECTIVE (Helikar et al., 2012); GINSIM (Naldi et al., 2018); BIOLQM (Naldi, in review[6]); MABOSS (Stoll et al., 2017); PINT (Paulevé, 2017); NUSMV (Cimatti et al., 2002); BOOLSIM (Garg et al., 2008); BOOLEANNET (Albert et al., 2008); PYBOOLNET (Klarner et al., 2017); BOOLNET (Müssel et al., 2010); CELLNOPT (Terfve et al., 2012); CASPOTS (Ostrowski et al., 2016). The current CoLoMoTo Docker (2018-03-31) ships the software indicated with a bold font and a light blue background. "Model repository" refers to searchable databases of models; "Model edition" refers to the features related to creating and modifying a qualitative model, where "*ab initio*" refers to the interactive model building from scratch, and "transformations" refers to operations such as mutations, Booleanization, model reduction, etc. "Topological analysis" refers to the extraction of features from the regulatory graph, such as the different feedback cycles, graph theory measures, etc. "Dynamical analysis" refers to properties related to the state transition graph of Boolean/multi-valued networks, where "Attractor analysis" refers to the identification of stable states, cyclic attractors, and related features; "Simulation" refers to the sampling of trajectories within the state transition graph, possibly parameterized with stochastic rates and mutations; "Formal verification and control" refers to exhaustive analyses for assessing strictly temporal properties, such as reachability, and deducing mutations for controlling the system. Finally, "Model inference" refers to the derivation of Boolean/multi-valued network which are compatible with given properties and observation data.

and can be accessed by standard workflow systems, such as SNAKEMAKE (Köster and Rahmann, 2012).

Then, we developed a collection of Python modules to provide a unified interface to the features of the selected software tools. The Python modules allow to parameterize and execute the different analyses, and fetch their results, which can then be further processed, including by a different tool through its respective Python module. This uniform Python interface is particularly relevant in the Jupyter web interface (Ragan-Kelley et al., 2014), where it allows editing executable notebooks on qualitative biological networks by seamlessly combining different software tools.

3.1. The CoLoMoTo Docker Image

Overall, we witness a growing ecosystem of software tools based on different technologies and offering a wide range of complementary features. Noteworthy, these tools typically rely on tailored formalisms and settings, which enable specific methods but at the same time affect the results. One obvious example is the consideration of a specific updating mode, as synchronous and asynchronous dynamics may differ extensively. Furthermore, to address increasingly large networks, many tools rely on advanced data-structures and resolution methods, which are implemented in dedicated software libraries. The distribution of these tools then become challenging, as they rely on numerous dependencies, often difficult to install or available only for a specific operating systems (most of the time GNU/Linux).

The Docker container technology allows to circumvent such distribution issues by providing a mean to supply pre-installed and fully configured software environments in so-called *Docker images*. On GNU/Linux, the execution of a Docker image consists mainly in executing the software in an isolated environment, requiring no operating system

virtualization. Therefore, the overhead of using Docker on GNU/Linux is close to zero. A Docker image can also be executed on macOS or Microsoft Windows without any modification. On these operating systems, Docker relies on virtualization technologies, which are relatatively lightweight and result in limited performance loss on recent hardware.

The current CoLoMoTo Docker image `colomoto/colomoto-docker:2018-03-31` contains the following pre-installed software for the logical modeling and analsyis of biological networks: GINsim (Naldi et al., 2018), bioLQM (Naldi, in review)[6], CellCollective (Helikar et al., 2012), MaBoSS (Stoll et al., 2017), Pint (Paulevé, 2017), and NuSMV (Cimatti et al., 2002). The CoLoMoTo Docker image then provides access to these tools without requiring any installation step beside installing Docker[7]. For instance, the Docker image can be used in association with a workflow manager to chain and run a series of software functionalities. Supplementary File "*SnakeMake*" provides an example of SnakeMake workflow relying on GINsim and NuSMV.

An important challenge is the maintenance and extendibility of such Docker images to reduce the complexity of upgrading or adding software tools with their respective dependencies. To that aim, we require that each software tool is independently packaged for GNU/Linux using the *Conda package manager*[8]. We then rely on the dependency management system of Conda to ensure that the correct pre-requisites are installed in the Docker image[9]. A beneficial side effect of this technical choice is that the aforementioned software tools can be installed on GNU/Linux platforms using Conda, without using Docker.

3.2. A Unified Interface for Calling and Chaining Tools With Python

The software tools considered for the CoLoMoTo Docker image present different interfaces: CellCollective is a web application, GINsim has a graphical user interface along with a scripting interface, bioLQM has a command line and a scripting interface, Pint has a command line and a Python interface, MaBoSS has a command line interface. GINsim, CellCollective and bioLQM support the SBML-qual format, while bioLQM provides the conversion of a standard SBML-qual model into Pint or MaBoSS model formats, thereby enabling the exchange of models between all these tools.

The recourse to different interfaces complicates the design of a model analysis combining multiple tools. To address this issue, we have developed a Python interface for each of the tools embedded in the CoLoMoTo Docker image, which greatly ease the execution of different tool functionalities, fetch the results, and use these as input for other executions.

Each tool comes with a dedicated Python module, providing a set of functions to invoke the underlying software tool appropriately. Therefore, from a single Python shell, one can invoke and chain analyses performed by different tools. This can be seen as an improved command line interface, greatly enhanced by the use of intermediate Python objects. Such an approach also promotes the use of standard Python data-structures to store objects such as model states or graphs, which can then be processed by common Python libraries, e.g., Pandas[10] or NetworkX[11].

Hereafter, we give an overview of the resulting Python programming interface, focusing on the general model input mechanism and the main features implemented for each of the software tools.

TABLE 1 | Model input formats for the software tools included in the CoLoMoTo Docker image.

Software tool	Supported input formats
bioLQM	SBML-qual (`.sbml`), raw logical functions, truth table
GINsim	GINML (`.ginml`, `.zginml`)
Pint	Automata network (`.an`)
MaBoSS	Dedicated network/configuration files (`.bnd`/`.cfg`)
NuSMV	SMV file (`.smv`)

3.2.1. Model Input and Tool Conversions

Despite their very different features, all the tools considered here take as input a logical model, in an adequate format. All the related Python modules provide a `load` function, which takes as input the location of the model, being a local file, for instance:

```
m = biolqm.load("path/to/localfile.sbml")
```

a web link to a file, as obtained on GINsim repository[12] for instance:

```
m = biolqm.load("http://ginsim.org/sites/default/
files/Traynard_Boolean_MamCC_Apr2016.sbml")
```

or a web link to the model on CellCollective, for instance:

```
m = biolqm.load("https://cellcollective.org/
#5128/lac-operon")
```

In each case, the returned object (identified by `m` in the above examples) is a Python object representing the loaded model and defined specifically for the corresponding tool (Python module). **Table 1** lists the supported input format for each software tool.

When possible, Python modules provide functions to convert a model for a compatible tool. These functions are of the form `moduleA.to_moduleB(modelA)`. **Figure 2** lists the currently supported model conversions. The following Python code shows an example of usage:

```
lrg = ginsim.load("http://ginsim.org/sites/default/
files/Traynard_Boolean_MamCC_Apr2016.sbml")
lqm = ginsim.to_biolqm(lrg)
an = biolqm.to_pint(lqm)
```

Here, `lrg` is a Python object representing a GINsim model, `lqm` is a Python object representing a bioLQM model, and `an` is a Python object representing a Pint model.

[7] See https://docker.com for installation instructions.
[8] https://conda.io
[9] CoLoMoTo-related conda packages are available in the *colomoto* conda channel. See https://anaconda.org/colomoto
[10] https://pandas.pydata.org
[11] http://networkx.github.io
[12] http://ginsim.org/models_repository

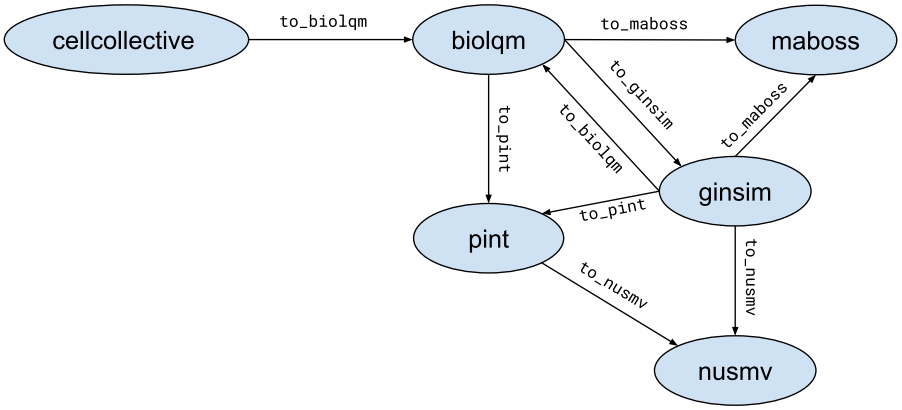


FIGURE 2 | Supported model conversions between Python modules.

3.2.2. CellCollective – Modeling Platform, Repository, and Knowledge Base

The `cellcollective` Python module allows connecting to the CellCollective (Helikar et al., 2012) web application (https://www.cellcollective.org), in order to download the models in SBML-qual format and extract network node meta-data (e.g., UnitProt identifiers) when available. The Supplementary File "*Notebooks/demo-cellcollective*"[13] provides a brief demonstration of the Python module usage.

3.2.3. GINsim – Regulatory Network Modeling

The `ginsim` Python module provides direct access to the Java programming interface of GINsim (Naldi et al., 2018). GINsim is available and documented at http://www.ginsim.org. In particular, besides the export of a GINsim model into various file formats, the Python module allows to visualize the network regulatory graph, with the activation and inhibition relationships between the nodes. The visualization function (`ginsim.show`) optionally takes as argument a Python dictionary associating a level with each node; then, the nodes of the network are colored according to these levels. This is illustrated in the Supplementary File "*Notebooks/demo-ginsim*"[14].

3.2.4. bioLQM – Qualitative Model Toolbox

The `biolqm` Python module provides direct access to the Java programming interface of bioLQM (Naldi, in review[6]). bioLQM is available and documented at http://colomoto.org/biolqm. bioLQM supports the conversion of SBML-qual files, GINML files, as well as simple textual files specifying the raw logical functions into the formats associated with the different software tools. Besides the file format features, bioLQM implements *model modifications*, such as mutations forcing the value of given nodes, iterative model reduction (see above), model reversal, the conversion of multi-valued model into Boolean ones, as well as the computation of *stable states*, *trap spaces*, and *simulations*. Part of these features are illustrated in the Supplementary File "*Notebooks/demo-biolqm*"[15].

3.2.5. Pint – Formal Predictions for Controlling Trajectories

The `pypint` Python module provides complete access to features documented at https://loicpauleve.name/pint. The software Pint is devoted to the analysis of trajectories in very large-scale asynchronous Boolean and multi-valued networks (Paulevé, 2017). Its main features include the verification of the existence of a trajectory reaching a state of interest (*reachability*), the identification of common points between *all* the trajectories leading to a state of interest (*cut sets*), and the *formal prediction of mutations* preventing the existence of any trajectory to the given state. These features are illustrated in the Supplementary File "*Notebooks/demo-pint*"[16].

3.2.6. NuSMV – Model Verification

The `nusmv` Python module provides a simple interface to the NuSMV model checker for verifying LTL (trace) and CTL (computation tree) temporal logic properties (Cimatti et al., 2002) . The specification of LTL and CTL properties can be facilitated using the `colomoto.temporal_logics` Python module, which takes advantage of Python objects for the different logical operators and ease their combination.

Let us consider the following example using the CTL operators from the aforementioned Python module:

```
p1 = AG (S(a=1))
p2 = EF (p1)
```

[13]The notebook can be previewed and downloaded at https://nbviewer.jupyter.org/github/colomoto/colomoto-docker/blob/2018-03-31/tutorials/CellCollective/CellCollective%20-%20Knowledge%20Base.ipynb

[14]The notebook can be previewed and downloaded at https://nbviewer.jupyter.org/github/colomoto/colomoto-docker/blob/2018-03-31/tutorials/GINsim/GINsim%20-%20visualization.ipynb

[15]The notebook can be previewed and downloaded at https://nbviewer.jupyter.org/github/colomoto/colomoto-docker/blob/2018-03-31/tutorials/bioLQM/bioLQM_tutorial.ipynb

[16]The notebook can be previewed and downloaded at https://nbviewer.jupyter.org/github/colomoto/colomoto-docker/blob/2018-03-31/tutorials/Pint/quick-tutorial.ipynb

Here, the variable `p1` is a CTL formula specifying that the node *a* is active (`S(a=1)`) in all the reachable states (`AG` operator). The variable `p2` is a CTL formula specifying that there exists a trajectory leading to a state (`EF` operator) from which the property `p1` is verified.

In the above example, `S` specifies a property on a state, by giving the values of some nodes of the network. The conversion of a network model into NuSMV format depends on the tool used, sometimes introducing different variable names for the nodes of the original biological network. But this technical point is transparent for the user: the `nusmv` Python module will automatically translate the node names into the correct NuSMV variable names.

The Supplementary File "*Notebooks/demo-nusmv*"[17] gives a simple example of usage of the `nusmv` Python module to verify properties of a GINsim model.

3.2.7. MaBoSS – Stochastic Simulations

The `maboss` Python module provides an interface to MaBoSS, available at https://maboss.curie.fr, as well as basic plotting functionalities (Stoll et al., 2017). The purpose of MaBoSS is to perform stochastic simulations of a Boolean network, where the propensity of transitions (probabilistic rates) are explicitly specified. The Python module allows to fully define and parameterize a model, as well as to parse an existing MaBoSS model and modify it programmatically. The object returned after the simulations can then be used to plot the probability of node activation over time, and the proportion of states in which the simulations ended, in order to estimate the probability of reaching different attractors. The Supplementary File "*Notebooks/demo-maboss*"[18] provides a brief tutorial to the main features of the `maboss` Python module.

3.2.8. Advanced Combinations of Tools

These Python modules provide a unified interface to chain different tools and process their results. The small tutorials referenced above show simple chaining of tools, most of the time using a tool to import a model (e.g., from CellCollective or GINsim) and convert it (using bioLQM) for specific analysis by another tool. As the Python functions of the different modules rely on standard Python data-structures, such as lists and dictionaries, it is possible to easily re-use the result from a tool function as input to the function of a different tool. A simple example is provided in Supplementary File "*Notebooks/demo-ginsim*"[19], where we use bioLQM to compute the stable states of a GINsim model, and then give one of the resulting state as input to GINsim `show` function to display it over the regulatory graph.

Moreover, one can use the programmatic features of Python to implement advanced algorithms for executing multiple analyses and process their results. For instance, one can program loops to iterate over a list of results of a preceding analysis from one tool to perform a subsequent analysis on each result with another tool. This is illustrated in the Supplementary File "*Notebooks/demo-pint+maboss*"[20], where we use Pint to formally predict combinations of mutations controlling the existence of trajectories toward a specified state; then, we quantify with MaBoSS the efficiency of applying only partially the predicted combinations, by evaluating each related double-mutants. The example involves Python `for` loops and a function to enumerate all possible subsets provided by the standard Python library. The notebook also relies on CellCollective to fetch the model, and on bioLQM to perform the adequate model conversions.

3.3. CoLoMoTo Jupyter Interactive Notebook

Jupyter[21] is a software providing an interactive web interface for creating documents, called *notebooks*, mixing code, equations, and formatted texts. A notebook typically describes a full analysis workflow, combining textual explanations, the code itself, along with parameters to reproduce the results. A notebook is a single file, which can be easily modified, shared, re-executed, and visualized online. The short tutorials of the previous section provided in the Supplementary File "*Notebooks*" are actually Jupyter notebooks (files with the extension `.ipynb`) and can be re-executed using Jupyter.

A Jupyter notebook is made of a sequence of so-called *cells*, which can contain formatted text, including sections, links, images, tables, etc., or which can contain code in a specified programming language, typically Python. A code cell can be executed (by pressing Shift-Enter) and the value returned by the code is displayed below the cell. The display format is selected according to the type of the returned value (image, graph, list, table, ...) to offer an adequate visualization.

Having a unified Python interface to invoke the CoLoMoTo software tools, one can directly create Jupyter notebooks for the analysis of qualitative biological networks using these tools, as shown in Supplementary File "*Notebooks*" and in the companion publication providing a complete model analysis workflow (Levy et al., 2018).

We added several features in the CoLoMoTo Python modules to increase interactivity and improve the user experience for editing Jupyter notebooks. First, menus provide pre-defined Python code for accessing to the main features of the tools. **Figure 3** shows a screenshot during the edition of a Jupyter notebook with its graphical interface. Next, we added the possibility to interactively upload a model file. This feature is particularly useful when used in combination with Docker, or on a remote server with no direct access to the user file system. Finally, some Python modules, in particular the `maboss` Python

[17] The notebook can be previewed and downloaded at https://nbviewer.jupyter.org/github/colomoto/colomoto-docker/blob/2018-03-31/tutorials/NuSMV/NuSMV%20with%20GINsim.ipynb

[18] The notebook can be previewed and downloaded at https://nbviewer.jupyter.org/github/colomoto/colomoto-docker/blob/2018-03-31/tutorials/MaBoSS/MaBoSS%20-%20Quick%20tutorial.ipynb

[19] The notebook can be previewed and downloaded at https://nbviewer.jupyter.org/github/colomoto/colomoto-docker/blob/2018-03-31/tutorials/GINsim/GINsim%20-%20visualization.ipynb

[20] The notebook can be previewed and downloaded at https://nbviewer.jupyter.org/github/colomoto/colomoto-docker/blob/2018-03-31/tutorials/MaBoSS/Predict%20mutations%20with%20Pint,%20refine%20with%20MaBoSS.ipynb

[21] http://jupyter.org

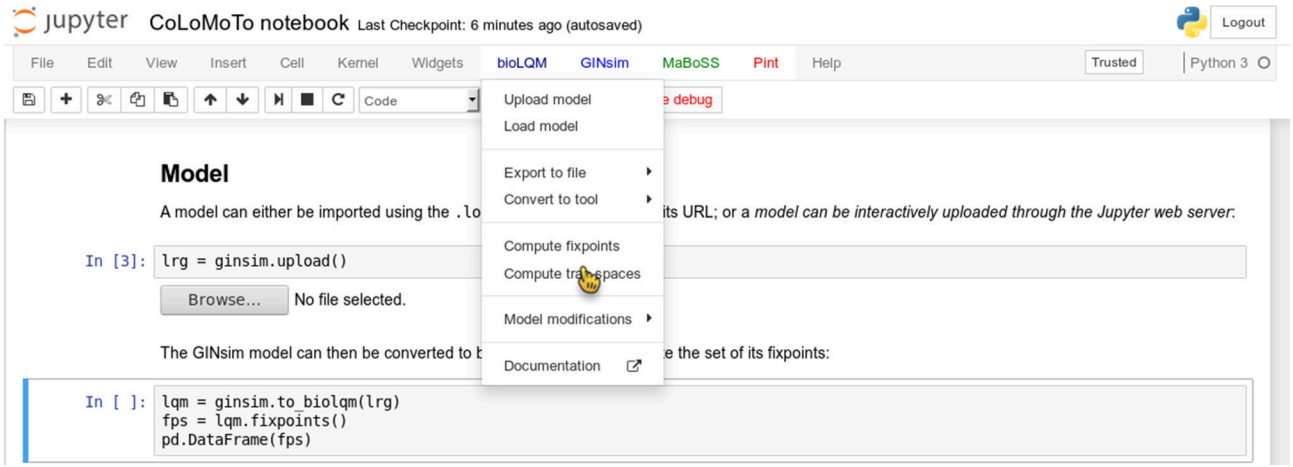


FIGURE 3 | Screenshot during a Jupyter notebook edition showing the menu of the BIOLQM tool.

module, provide JavaScript widgets to generate Python code interactively.

The Jupyter notebook server is included in the CoLoMoTo Docker image (see the Discussion section for a quick usage guide), while a public demonstration web instance is available at http://tmpnb.colomoto.org.

4. REPRODUCIBILITY OF COMPUTATIONAL ANALYSES

4.1. From Repeatability to Reproducibility

The literature provides a range of definitions for the reproducibility of *in silico* experiments by analogy to wet lab experiments (Drummond, 2009; Freire et al., 2012; Stodden et al., 2013; Freire et al., 2016; Goodman et al., 2016; Lewis et al., 2016; Cohen-Boulakia et al., 2017). Four *levels* of reproducibility are commonly distinguished.

An *in silico* experiment is said to be *repeated* when it is performed using the same computational set-up as the original experiment. The major goal of the *repeat* task is to check whether the initial experimental result was correct and can be obtained again. The difficulty lies in recording as much information as possible to repeat the experiment so that the same conclusion can be drawn. Interestingly, Freire et al. (2012) discusses the granularity at which information (experiments, data sets, parameters, environment) should or could be recorded, and underlines the fact that the key point is to determine the right balance between the effort required to record information and the capability of obtaining identical results.

An *in silico* experiment is said to be *replicated* when it is performed in a new setting and computational environment, although similar to the original ones. When it can be successfully replicated, a result has a high level of robustness: it remained valid when using a similar (although different) protocol. A continuum of situations can be considered between repeated and replicated experiments.

A result is then defined as *reproduced*, in the broadest possible sense of the term, by denoting the situation where an experiment is performed within a different environment, with the aim to validate the same scientific hypothesis. In other words, what matters here is the conclusion obtained and not the methodology considered reaching it. Completely different approaches can be designed, different data sets can be used, as long as the experiments support the same scientific conclusion. A reproducible result is thus a high-quality result, confirmed in various ways.

A last important concept related to reproducibility is that of *reuse*, which denotes the case where a *different* experiment is performed, with similarities with an original experiment. A specific kind of reuse occurs when a single experiment is reused in a new context (and thus adapted to new needs), the experiment is then said to be *repurposed*.

It is worth noticing that *repeating* and *replicating* may appear to be technical challenges compared to *reproducing* and *reusing*, which are the most important scientific objectives. However, before investigating alternative ways of obtaining a result (to reach reproducibility), or before reusing a given methodology in a new context (to reach reuse), the original experiment has to be carefully tested, especially by reviewers or any peer, demonstrating its ability to be at least repeated and hopefully replicated (Freire et al., 2012; Stodden et al., 2014).

4.2. Repeat Analysis in the Same Software Environment

Ensuring that a sequence of computational analyses can be repeated by other scientists several months or years after its publication is difficult. Indeed, besides software availability, the version of the tools can be crucial: a new version of a tool can change the default parameters, and even some features, so that the published instructions become obsolete. Whereas a Docker image addresses efficiently the issue of making software available, providing a safe way for repeating a notebook

content years after its creation requires additional technical procedures.

First, CoLoMoTo Docker images are constructed by specifying explicitly the version of each software. Furthermore, an automatic validation procedure is performed by checking that a set of notebooks still execute without error. Once validated, the Docker image is then tagged with a time-stamp, typically the date of the image validation (of the form `YYYY-MM-DD`, e.g., 2018-03-31). These tagged images are then stored in the public Docker image registry, and can be retrieved any time later. The list of existing tags of `colomoto/colomoto-docker` Docker images can be viewed at https://hub.docker.com/r/colomoto/colomoto-docker/tags/.

When sharing a notebook, and notably when attaching it to a publication, it is highly recommended to specify the time-stamp of the Docker image in which the notebook has been executed. Then, by downloading the image with this specific tag, other users are ensured to repeat the execution in the exact same software environment. To help following this recommendation, we took two technical decisions. First, we do not use the default non-persistent tag for Docker images (*latest*). It means that the user has always to specify explicitly the time-stamp of the CoLoMoTo Docker image. To remove the burden of actively checking the list of existing time-stamps, we provide a script which, by default, fetches the most recent Docker image (see section 5). Second, when loading a CoLoMoTo-related Python module within a Docker container, a textual message indicating the time-stamp of the Docker image is displayed. Therefore, when created within a CoLoMoTo Docker image, notebooks always contain the required information to repeat their execution.

Because a Jupyter notebook is a single file containing everything to execute it, one can easily check if it can be *replicated* in a different software environment, e.g., using a more recent CoLoMoTo Docker image. Moreover, a notebook can be easily *repurposed* by modifying some arguments of the Python function calls, for instance changing the input model or analysis parameters. One can even define interactive notebooks describing a common model analysis, so that the user only needs to provide the input model and execute the Jupyter code cells, as shown in the Supplementary File "*Notebooks/demo-interactive-fixpoints*"[22] for the computation and visualization of the stable states of a bioLQM model.

4.3. Reproduce Analysis With a Different Method

Reproducing the same analysis with two different methods is a good mean to increase confidence in the results, as it reduces the chance of software misuse or that the results are affected by a software bug.

The subset of software tools selected for this first CoLoMoTo Docker image presentation already provides redundant implementations of equivalent model analyses, in particular for the identification of stable states and for the verification of temporal properties with NuSMV. To help switch between two tools for performing the same task, we harmonized the usage of Python module functions to ensure that the same functions with the same arguments generate equivalent results with different tools.

4.3.1. Stable States

There exists several methods to compute the full set of stable states (or fixed points) of a logical model, relying on different data-structures and different algorithms. The software BIOLQM implements the computation of stable states for Boolean and multi-valued logical models using a Java implementation of decision diagrams. In contrast, the software PINT implements the computation of stable states of Automata networks (a generalization of logical networks) using Boolean satisfaction constraints. As BIOLQM provides a conversion of Boolean/multi-valued network into equivalent Automata networks, it is possible to compute the stable states of a model with both software tools.

Both `biolqm` and `pypint` Python modules provide a `fixpoint` function taking as input the model instance of the corresponding tool and returning a list of Python dictionaries describing the stable states. Provided `lqm` is a BIOLQM model, the following Python code compute its stable states with both tools:

```
fps_biolqm = biolqm.fixpoints(lqm)
fps_pint = pypint.fixpoints(biolqm.to_pint(lqm))
```

The Supplementary File "*Notebooks/demo-reproducibility-fixpoints*"[23] shows a complete example of reproduction of stable state computation using BIOLQM and PINT.

4.3.2. Temporal Property Verification (Model-Checking)

Both GINSIM and PINT allow to export their respective model into NUSMV format, where temporal properties can be specified using LTL or CTL (see section 2.2). However, the generated NUSMV models have different features as the input formalisms of these tools rely on different paradigms: the specification is centered on logical rules in the case of Boolean/multi-valued networks in GINSIM, and on transitions (à la Petri nets) in the case of Automata networks in PINT. Nevertheless, in the appropriate settings, the verification of an equivalent CTL or LTL property should give the same result. Hence, the functions `ginsim.to_nusmv` and `pypint.to_nusmv` are implemented in such ways that, when using their default options, the resulting NUSMV models, albeit different, should produce identical results for identical temporal logic properties. Note, however, that each tool provides specific options for the NUSMV export, which can lead to incomparable results.

The following Python code uses operators defined in the Python module `colomoto.temporal_logics` to specify a

[22]The notebook can be previewed and downloaded at https://nbviewer.jupyter.org/github/colomoto/colomoto-docker/blob/2018-03-31/tutorials/bioLQM/Fixpoints%20(interactive).ipynb

[23]The notebook can be previewed and downloaded at https://nbviewer.jupyter.org/github/colomoto/colomoto-docker/blob/2018-03-31/tutorials/Reproducibility%20-%20fixpoints.ipynb

property `p`, meaning that from any state, there always exists a trajectory leading to a cyclic attractor where the level of node *a* can always oscillate. Then, assuming `lrg` is a GINSIM model, the code uses GINSIM and PINT conversions to NUSMV to perform model verification.

```
p = EF (AG (EF (S(a=0)) & EF (S(a=1))))

nusmv_ginsim = ginsim.to_nusmv(lrg)
nusmv_ginsim.add_ctl(p)
nusmv_ginsim.verify()

nusmv_pint = pypint.to_nusmv(ginsim.to_pint(lrg))
nusmv_pint.add_ctl(p)
nusmv_pint.verify()
```

Note that the Python object `p` represents the CTL property to be tested, whatever the origin of the model (GINSIM or PINT).

The Supplementary File "*Notebooks/demo-reproducibility-modelchecking*"[24] provides a more detailed example of the reproduction of model-checking results using GINSIM and PINT.

5. QUICK-USAGE GUIDE

On GNU/Linux, macOS, or Microsoft Windows, provided that Docker and Python are installed, a helper script to run the CoLoMoTo Docker image and the embedded Jupyter notebook can be installed and upgraded from a terminal using the following command[25]:

```
pip install -U colomoto-docker
```

The Docker image and the Jupyter notebook interface can be started by executing the following command in a terminal[26]:

```
colomoto-docker
```

Without any argument, the command will use the most recent CoLoMoTo Docker image. To use the image with a specific tag, append the `-V` option (e.g., `colomoto-docker -V 2018-03-31`).

The execution of this command will open a web page with the Jupyter notebook interface, enabling loading and execution of notebooks. A new notebook can be created by using the "New/Python3" menu. In this environment, the user has access to all CoLoMoTo Python modules. A code cell is executed by typing "Shift+Enter." The menu and tool bar allow quick access to the main Jupyter functionalities.

Warning: by default, the files within the Docker container are isolated from the running host computer, and are deleted when stopping the container. To have access to the files of the current directory of the host computer, the option `--bind` can be used:

```
colomoto-docker --bind .
```

The container can later be stopped by pressing Ctrl+C keys in the terminal. See `colomoto-docker --help` for other options. Additional documentation for running the CoLoMoto Docker image can be found at http://colomoto.org/notebook.

6. DISCUSSION

6.1. Academic Use Cases

The prime aim of the CoLoMoTo Interactive Notebook is to foster the production of accessible and reproducible computational analysis of biological models, with a focus on qualitative models, including Boolean and multi-valued networks. As demonstrated in the Supplementary File "*SnakeMake*", the CoLoMoTo Docker image can also be used in standard workflow systems, such as SnakeMake, to lighten the burden of installing the different software tools and make them accessible on different operating systems.

A notebook issued from the CoLoMoTo Docker image gives some guarantees of repeatability, as it contains references to the persistent Docker image to re-execute code in the same software environment. Therefore, the notebook file (with `.ipynb` extension) can be distributed as a Supplementary File of the related scientific article, along with instruction to run the Docker image. The Jupyter interface further allows to export the notebook in a static HTML file, which could also be joined as a Supplementary File to provide a quick visualization. A notebook can also be distributed independently, for instance by publishing it on *Gist*[27] or *myExperiment*[28] (Goble et al., 2010), to follow download and potential updates. For instance, the tutorial notebook presented by Levy et al. (2018) is hosted at https://gist.github.com/pauleve/a86717b0ae8750440dd589f778db428f. Services like Zenodo[29] further provide persistent DOI links to notebook files.

The CoLoMoTo Interactive Notebook is also relevant for teaching purposes. With Jupyter, students can straightforwardly execute, modify, and extend a template notebook to learn methods for analysing models of biological networks. Docker is a standard technology often supported by local cloud infrastructures, which can therefore provide dedicated resources to execute remotely and privately the CoLoMoTo Jupyter web interface.

6.2. Extending the CoLoMoTo Interactive Notebook

The CoLoMoTo Docker image can be easily extended to include additional tools. The Docker architecture allows inheriting from an existing container, adding a new layer with additional executables. Contributions are welcome through GitHub[30]. Each software tool must be usable from the Jupyter interface and should be able to connect with at least one other tool already included. Furthermore, a demonstration notebook should be

[24]The notebook can be previewed and downloaded at https://nbviewer.jupyter.org/github/colomoto/colomoto-docker/blob/2018-03-31/tutorials/Reproducibility%20-%20model%20checking.ipynb

[25]You may have to use `pip3` instead of `pip` depending on your configuration.

[26]If using Docker Toolbox, the command should be executed within the Docker Terminal.

[27]https://gist.github.com

[28]https://www.myexperiment.org

[29]https://zenodo.org

[30]Guidelines available at https://github.com/colomoto/colomoto-docker/blob/master/CONTRIBUTING.md

TABLE 2 | List of notebook files in supplemental data "Notebooks" (**Data Sheet 2**) demonstrating some features of the CoLoMoTo Interactive Notebook.

Notebook file name	Software tools involved
demo-cellcollective	CellCollective, bioLQM
demo-ginsim	GINsim, bioLQM
demo-biolqm	bioLQM
demo-interactive-fixpoints	bioLQM
demo-pint	Pint
demo-nusmv	GINsim, NuSMV
demo-maboss	MaBoSS
demo-pint+maboss	CellCollective, bioLQM, Pint, MaBoSS
demo-reproducibility-fixpoints	GINsim, bioLQM, Pint
demo-reproducibility-modelchecking	GINsim, Pint

provided to illustrate the tool usage and how it can be combined with other tools.

Currently, all the embedded tools require an already defined model. Nevertheless, once loaded, a model can be subsequently modified from the Python interface (see tool feature matrix in **Figure 1**). We are currently considering the development of a programmatic interface for model definition *ab initio*. One of the main challenge is to provide a decent visualization of the programmatically-created model. A potential direction is to include a visual edition module in the Jupyter interface, which represents a substantial development effort.

The support for standard exchange formats is key to enable reproducibility of analyses with different tools. In that sense, BIOLQM plays an important role for the CoLoMoTo Interactive Notebook as it provides bridges between SBML-qual standard specifications and numerous software tools (**Figure 2**). The Tellurium Notebook system by Sauro et al. (in review)[31] offers support for SED-ML to help reproduce quantitative simulation of biological networks. Future work should consider bringing this feature for qualitative models as well, in order to better meet FAIR (Findability, Accessibility, Interoperability, and Reusability) recommendations (Wittig et al., 2017).

AUTHOR CONTRIBUTIONS

AN, CH, DT, and LP designed the main principles of the CoLoMoTo Interactive notebook and its distribution. AN, CH, NL, and LP implemented the necessary Python modules, their integration in the Jupyter interface, and the Docker image. AN and LP edited the notebook tutorials, while CH edited the SnakeMake workflow example. All authors contributed to the writing of the article under the supervision of DT and LP.

[31] Preprint on bioRxiv: https://doi.org/10.1101/239004

All authors reviewed the content of this article and agreed to endorse it.

FUNDING

DT and CH acknowledge support from the French Plan Cancer, in the context of the projects CoMET (2014–2017) and SYSTAIM (2015–2019). DT and AN acknowledge support from the French Agence Nationale pour la Recherche (ANR), in the context of the project SCAPIN [ANR-15-CE15-0006-01]. CC and PM acknowledge support from the Fundação para a Ciência e a Tecnologia, through grants PTDC/BEX-BCB/0772/2014 and PTDC/EEI-CTP/2914/2014. TH acknowledges support from the National Institutes of Health (#5R35GM119770-02). SC-B acknowledges support from CNRS (défi Mastodons). AZ and LC acknowledge support from COLOSYS project in EU ERACoSysMed programme. AZ acknowledges support by the Ministry of education and science of Russia (Project No. 14.Y26.31.0022). AZ, LC, and LP acknowledge support from ANR in the context of the project ANR-FNR project AlgoReCell [ANR-16-CE12-0034]. LP and SC-B acknowledge support from Paris Ile-de-France Region (DIM RFSI) and Labex DigiCosme [ANR-11-LABEX-0045-DIGICOSME] operated by ANR as part of the program Investissement d'Avenir Idex Paris-Saclay [ANR-11-IDEX-0003-02].

ACKNOWLEDGMENTS

The authors thank the attendees of the fourth CoLoMoTo meeting in Paris, July 2017, for the insightful discussions which led to designing the CoLoMoTo Interactive notebook. The authors thank Laurent Darré and the technical staff at LRI, Université Paris-Sud, France, for providing the computing resources for hosting http://tmpnb.colomoto.org.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphys.2018.00680/full#supplementary-material

Data Sheet 1 | The supplemental file "SnakeMake" contains an example of SnakeMake workflow that uses the CoLoMoTo Docker image to execute complementary analyses.

Data Sheet 2 | The supplemental data "Notebooks" contains several short Jupyter notebooks which demonstrate different usage of the CoLoMoTo interactive notebook, listed in **Table 2**. The .ipynb files can be imported and executed within the Jupyter interface of the CoLoMoTo notebook, using the Docker image colomoto/colomoto-docker:2018-03-31. For each of these notebooks, a static HTML file previews the Jupyter rendering of the notebook, without any requirement. These notebooks can also be previewed and downloaded at https://nbviewer.jupyter.org/github/colomoto/colomoto-docker/tree/2018-03-31/tutorials.

REFERENCES

Abou-Jaoudé, W., Monteiro, P. T., Naldi, A., Grandclaudon, M., Soumelis, V., Chaouiya, C., et al. (2015). Model checking to assess t-helper cell plasticity. *Front. Bioeng. Biotechnol.* 2:86. doi: 10.3389/fbioe.2014.00086

Abou-Jaoudé, W., Traynard, P., Monteiro, P. T., Saez-Rodriguez, J., Helikar, T., Thieffry, D., et al. (2016). Logical modeling and dynamical analysis of cellular networks. *Front. Genet.* 7:94. doi: 10.3389/fgene.2016.00094

Albert, I., Thakar, J., Li, S., Zhang, R., and Albert, R. (2008). Boolean network simulations for life scientists. *Source Code Biol. Med.* 3:16. doi: 10.1186/1751-0473-3-16

Baker, M. (2016). 1,500 scientists lift the lid on reproducibility. *Nat. News* 533:452. doi: 10.1038/533452a

Bartocci, E. and Lió, P. (2016). Computational modeling, formal analysis, and tools for systems biology. *PLOS Comput. Biol.* 12:e1004591. doi: 10.1371/journal.pcbi.1004591

Batt, G., Ropers, D., de Jong, H., Geiselmann, J., Mateescu, R., Page, M., et al. (2005). Validation of qualitative models of genetic regulatory networks by model checking: analysis of the nutritional stress response in *Escherichia coli*. *Bioinformatics* 21(Suppl. 1), i19–i28. doi: 10.1093/bioinformatics/bti1048

Begley, C. G., and Ellis, L. M. (2012). Drug development: raise standards for preclinical cancer research. *Nature* 483, 531–533. doi: 10.1038/483531a

Begley, C. G., and Ioannidis, J. P. (2015). Reproducibility in science improving the standard for basic and preclinical research. *Circ. Res.* 116, 116–126. doi: 10.1161/CIRCRESAHA.114.303819

Chaouiya, C., Bérenguier, D., Keating, S. M., Naldi, A., van Iersel, M. P., Rodriguez, N., et al. (2013). SBML qualitative models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools. *BMC Syst. Biol.* 7:135. doi: 10.1186/1752-0509-7-135

Chaouiya, C., Keating, S. M., Berenguier, D., Naldi, A., Thieffry, D., van Iersel, M. P., et al. (2015). The Systems Biology Markup Language (SBML) level 3 package: qualitative models, version 1, release 1. *J. Integr. Bioinform.* 12:270. doi: 10.1515/jib-2015-270

Cimatti, A., Clarke, E., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., et al. (2002). "NuSMV Version 2: an OpenSource Tool for Symbolic Model Checking," in *Proceedings of International Conference on Computer-Aided Verification (CAV 2002), Vol. 2404 of LNCS* (Copenhagen: Springer). doi: 10.1007/3-540-45657-0_29

Cohen-Boulakia, S., Belhajjame, K., Collin, O., Chopard, J., Froidevaux, C., Gaignard, A., et al. (2017). Scientific workflows for computational reproducibility in the life sciences: status, challenges and opportunities. *Fut. Gen. Comput. Syst.* 75, 284–298. doi: 10.1016/j.future.2017.01.012

Drummond, C. (2009). "Replicability is not reproducibility: nor is it good science," in *Proceedings of the Evaluation Methods for Machine Learning Workshop at the 26th ICML* (Montreal, QC).

Errington, T. M., Iorns, E., Gunn, W., Tan, F. E., Lomax, J., and Nosek, B. A. (2014). An open investigation of the reproducibility of cancer biology research. *Elife* 3:e04333. doi: 10.7554/eLife.04333

Fauré, A., Naldi, A., Chaouiya, C., and Thieffry, D. (2006). Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle. *Bioinformatics* 22, 124–31. doi: 10.1093/bioinformatics/btl210

Freire, J., Bonnet, P., and Shasha, D. (2012). "Computational reproducibility: state-of-the-art, challenges, and database research opportunities," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data* (Scottsdale, AZ), 593–596. doi: 10.4230/DagRep.6.1.108

Freire, J., Fuhr, N., and Rauber, A. (2016). "Reproducibility of data-oriented experiments in e-science," in *Dagstuhl Seminar 16041* (Dagstuhl), 108–159.

Gan, X., and Albert, R. (2018). General method to find the attractors of discrete dynamic models of biological systems. *Phys. Rev. E* 97:042308. doi: 10.1103/PhysRevE.97.042308

Garg, A., Di Cara, A., Xenarios, I., Mendoza, L., and De Micheli, G. (2008). Synchronous versus asynchronous modeling of gene regulatory networks. *Bioinformatics* 24, 1917–1925. doi: 10.1093/bioinformatics/btn336

Goble, C. A., Bhagat, J., Aleksejevs, S., Cruickshank, D., Michaelides, D., Newman, D., et al. (2010). myExperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucleic Acids Res.* 38(Suppl. 2), W677–W682. doi: 10.1093/nar/gkq429

Goodman, S. N., Fanelli, D., and Ioannidis, J. P. (2016). What does research reproducibility mean? *Sci. Transl. Med.* 8:341ps12. doi: 10.1126/scitranslmed.aaf5027

Helikar, T., Kowal, B., McClenathan, S., Bruckner, M., Rowley, T., Madrahimov, A., et al. (2012). The Cell Collective: toward an open and collaborative approach to systems biology. *BMC Syst. Biol.* 6:96. doi: 10.1186/1752-0509-6-96

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., et al. (2003). The Systems Biology Markup Language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* 19, 524–531. doi: 10.1093/bioinformatics/btg015

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.* 22, 437–467. doi: 10.1016/0022-5193(69)9001

Klarner, H., Streck, A., and Siebert, H. (2017). PyBoolNet: a python package for the generation, analysis and visualization of Boolean networks. *Bioinformatics* 33, 770–772. doi: 10.1093/bioinformatics/btw682

Köster, J., and Rahmann, S. (2012). Snakemake - a scalable bioinformatics workflow engine. *Bioinformatics* 28, 2520–2522. doi: 10.1093/bioinformatics/bts480

Le Novère, N., Finney, A., Hucka, M., Bhalla, U. S., Campagne, F., Collado-Vides, J., et al. (2005). Minimum information requested in the annotation of biochemical models (MIRIAM). *Nat. Biotechnol.* 23, 1509–1515. doi: 10.1038/nbt1156

Levy, N., Naldi, A., Hernandez, C., Stoll, G., Thieffry, D., Zinovyev, A., et al. (2018). Prediction of mutations to control pathways enabling tumour cell invasion with the CoLoMoTo interactive notebook (tutorial). *Front. Physiol.* 9:787. doi: 10.3389/fphys.2018.00787

Lewis, J., Breeze, C. E., Charlesworth, J., MacLaren, O. J., and Cooper, J. (2016). Where next for the reproducibility agenda in computational biology? *BMC Syst. Biol.* 10:52. doi: 10.1186/s12918-016-0288-x

Müssel, C., Hopfensitz, M., and Kestler, H. (2010). BoolNet–an R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics* 26, 1378–1380. doi: 10.1093/bioinformatics/btq124

Naldi, A., Hernandez, C., Abou-Jaoudé, W., Monteiro, P. T., Chaouiya, C., and Thieffry, D. (2018). Logical modelling and analysis of cellular regulatory networks with GINsim 3.0. *Front. Physiol.* 9:646. doi: 10.3389/fphys.2018.00646

Naldi, A., Monteiro, P. T., Müssel, C., Consortium for Logical Models and Tools, Kestler, H. A., Thieffry, D., Xenarios, I., et al. (2015). Cooperative development of logical modelling standards and tools with CoLoMoTo. *Bioinformatics* 31, 1154–1159. doi: 10.1093/bioinformatics/btv013

Naldi, A., Remy, E., Thieffry, D., and Chaouiya, C. (2011). Dynamically consistent reduction of logical regulatory graphs. *Theor. Comput. Sci.* 412, 2207–2218. doi: 10.1016/j.tcs.2010.10.021

Ostrowski, M., Paulevé, L., Schaub, T., Siegel, A., and Guziolowski, C. (2016). Boolean network identification from perturbation time series data combining dynamics abstraction and logic programming. *Biosystems* 149, 139–153. doi: 10.1016/j.biosystems.2016.07.009

Paulevé, L. (2017). "Pint: a static analyzer for transient dynamics of qualitative networks with IPython interface," in *CMSB 2017 - 15th Conference on Computational Methods for Systems Biology Volume 10545 of Lecture Notes in Computer Science* (Darmstadt: Springer), 370–316. doi: 10.1007/978-3-319-67471-1_20

Paulevé, L. (in press). Reduction of qualitative models of biological networks for transient dynamics analysis. *IEEE/ACM Trans. Comput. Biol. Bioinform.* doi: 10.1109/TCBB.2017.2749225

Peng, R. D. (2009). Reproducible research and biostatistics. *Biostatistics* 10, 405–408. doi: 10.1093/biostatistics/kxp014

Ragan-Kelley, M., Perez, F., Granger, B., Kluyver, T., Ivanov, P., Frederic, J., et al. (2014). "The Jupyter/IPython architecture: a unified view of computational research, from interactive exploration to communication and publication," in *AGU Fall Meeting Abstracts* (San Francisco, CA).

Richter, S. H., Garner, J. P., Auer, C., Kunert, J., and Würbel, H. (2010). Systematic variation improves reproducibility of animal experiments. *Nat. Methods* 7, 167–168. doi: 10.1038/nmeth0310-167

Santori, G. (2016). Journals should drive data reproducibility. *Nature* 535, 355–355. doi: 10.1038/535355b

Siebert, H. (2009). Deriving behavior of boolean bioregulatory networks from subnetwork dynamics. *Math. Comput. Sci.* 2, 421–442. doi: 10.1007/s11786-008-0064-4

Smith, M. A., and Houghton, P. (2013). A proposal regarding reporting of *in vitro* testing results. *Clin. Cancer Res.* 19, 2828–2833. doi: 10.1158/1078-0432.CCR-13-0043

Stodden, V., Guo, P., and Ma, Z. (2013). Toward reproducible computational research: an empirical analysis of data and code policy adoption by journals. *PLoS ONE* 8:e67111. doi: 10.1371/journal.pone.0067111

Stodden, V., Leisch, F., and Peng, R. D. (2014). *Implementing Reproducible Research*. CRC Press.

Stoll, G., Caron, B., Viara, E., Dugourd, A., Zinovyev, A., Naldi, A., et al. (2017). MaBoSS 2.0: an environment for stochastic Boolean modeling. *Bioinformatics* 33, 2226–2228. doi: 10.1093/bioinformatics/btx123

Stoll, G., Viara, E., Barillot, E., and Calzone, L. (2012). Continuous time boolean modeling for biological signaling: application of gillespie algorithm. *BMC Systems Biology* 6:116. doi: 10.1186/1752-0509-6-116

Talcott, C., and Dill, D. L. (2006). "Multiple representations of biological processes," in *Transactions on Computational Systems Biology VI*, eds C. Priami and G. Plotkin (Berlin; Heidelberg: Springer Science Business Media), 221–245. doi: 10.1007/11880646_10

Terfve, C., Cokelaer, T., Henriques, D., MacNamara, A., Goncalves, E., Morris, M. K., et al. (2012). CellNOptR: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms. *BMC Syst. Biol.* 6:133. doi: 10.1186/1752-0509-6-133

Thomas, R. (1973). Boolean formalization of genetic control circuits. *J. Theor. Biol.* 42, 563–585. doi: 10.1016/0022-5193(73)90247-6

Todd, R. G., and Helikar, T. (2012). Ergodic sets as cell phenotype of budding yeast cell cycle. *PLoS ONE* 7:e45780. doi: 10.1371/journal.pone.0045780

Traynard, P., Fauré, A., Fages, F., and Thieffry, D. (2016). Logical model specification aided by model-checking techniques: application to the mammalian cell cycle regulation. *Bioinformatics* 32, i772–i780. doi: 10.1093/bioinformatics/btw457

Veliz-Cuba, A., Aguilar, B., Hinkelmann, F., and Laubenbacher, R. (2014). Steady state analysis of boolean molecular network models via model reduction and computational algebra. *BMC Bioinformatics* 15:221. doi: 10.1186/1471-2105-15-221

Waltemath, D., Adams, R., Beard, D. A., Bergmann, F. T., Bhalla, U. S., Britten, R., et al. (2011a). Minimum Information About a Simulation Experiment (MIASE). *PLoS Comput. Biol.* 7:e1001122. doi: 10.1371/journal.pcbi.1001122

Waltemath, D., Adams, R., Bergmann, F. T., Hucka, M., Kolpakov, F., Miller, A. K., et al. (2011b). Reproducible computational biology experiments with SED-ML – the Simulation Experiment Description Markup Language. *BMC Syst. Biol.* 5:198. doi: 10.1186/1752-0509-5-198

Wittig, U., Rey, M., Weidemann, A., and Müller, W. (2017). Data management and data enrichment for systems biology projects. *J. Biotechnol.* 261, 229–237. doi: 10.1016/j.jbiotec.2017.06.007

Yaffe, M. B. (2015). Reproducibility in science. *Sci. Signal.* 8:eg5. doi: 10.1126/scisignal.aaa5764

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

PROTOCOLS
published: 06 July 2018
doi: 10.3389/fphys.2018.00787

Prediction of Mutations to Control Pathways Enabling Tumor Cell Invasion with the CoLoMoTo Interactive Notebook (Tutorial)

Nicolas Levy[1,2], *Aurélien Naldi*[3], *Céline Hernandez*[3], *Gautier Stoll*[4,5,6,7,8], *Denis Thieffry*[3], *Andrei Zinovyev*[9,10,11,12], *Laurence Calzone*[9,10,11] and *Loïc Paulevé*[1]*

[1] LRI UMR 8623, Centre National de la Recherche Scientifique, Université Paris-Sud, Université Paris-Saclay, Orsay, France, [2] École Normale Supérieure de Lyon, Lyon, France, [3] Computational Systems Biology Team, Institut de Biologie de l'École Normale Supérieure, Centre National de la Recherche Scientifique UMR8197, INSERM U1024, École Normale Supérieure, PSL Université, Paris, France, [4] Université Paris Descartes, Sorbonne Paris Cité, Paris, France, [5] Équipe 11 Labellisée Ligue Nationale contre le Cancer, Centre de Recherche des Cordeliers, Paris, France, [6] Institut National de la Santé et de la Recherche Médicale, Paris, France, [7] Université Pierre et Marie Curie, Paris, France, [8] Metabolomics and Cell Biology Platforms, Gustave Roussy Cancer Campus, Villejuif, France, [9] Institut Curie, PSL Research University, Paris, France, [10] INSERM U900, Paris, France, [11] MINES ParisTech, PSL Research University, CBIO-Centre for Computational Biology, Paris, France, [12] Lobachevsky University, Nizhni Novgorod, Russia

Boolean and multi-valued logical formalisms are increasingly used to model complex cellular networks. To ease the development and analysis of logical models, a series of software tools have been proposed, often with specific assets. However, combining these tools typically implies a series of cumbersome software installation and model conversion steps. In this respect, the *CoLoMoTo Interactive Notebook* provides a joint distribution of several logical modeling software tools, along with an interactive web Python interface easing the chaining of complementary analyses. Our computational workflow combines (1) the importation of a GINsim model and its display, (2) its format conversion using the Java library BioLQM, (3) the formal prediction of mutations using the OCaml software Pint, (4) the model checking using the C++ software NuSMV, (5) quantitative stochastic simulations using the C++ software MaBoSS, and (6) the visualization of results using the Python library matplotlib. To illustrate our approach, we use a recent Boolean model of the signaling network controlling tumor cell invasion and migration. Our model analysis culminates with the prediction of sets of mutations presumably involved in a metastatic phenotype.

Keywords: Boolean networks, stochastic simulations, model verification, software tools, reproducibility

OPEN ACCESS

Edited by:
Pierre De Meyts,
de Duve Institute, Belgium

Reviewed by:
Katsuhiko Murakami,
Fujitsu Laboratories, Japan
David Phillip Nickerson,
University of Auckland, New Zealand

***Correspondence:**
Loïc Paulevé
loic.pauleve@lri.fr

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: *05 April 2018*
Accepted: *06 June 2018*
Published: *06 July 2018*

Citation:
Levy N, Naldi A, Hernandez C, Stoll G, Thieffry D, Zinovyev A, Calzone L and Paulevé L (2018) Prediction of Mutations to Control Pathways Enabling Tumor Cell Invasion with the CoLoMoTo Interactive Notebook (Tutorial). Front. Physiol. 9:787. doi: 10.3389/fphys.2018.00787

1. INTRODUCTION

Boolean and multi-valued logical formalisms are increasingly used to model complex cellular networks (see e.g., Helikar et al., 2012; Zaudo and Albert, 2015; Collombet et al., 2017). A logical model is usually defined in three steps:

1) The delineation of a regulatory graph, where the vertices (nodes) represent signaling or regulatory components (proteins, genes, microRNAs, etc.), while the arcs (arrows) represent regulatory interactions between pairs of components. These arcs are labeled by a sign: positive in the case of activation, negative in the case of an inhibition (multiple arcs between two nodes may be considered but are not used here).

2) A discrete variable is associated with each node. In the simplest cases, as hereafter, these variables are Boolean, i.e., they can take only two values (0 or 1), denoting the absence/inactivity or the presence/activity of the corresponding components.
3) Finally, a logical rule is associated with each component to specify the combinations enabling its activation. More precisely, this rule combines the different variables corresponding to the regulatory components using the logical negation (denoted by `!`), conjunction (denoted by `&`) and disjunction (denoted by `|`). For example, the rule associated with the component GF in the model considered below is `!CDH1 & (GF | CDH2)`, which reads as "the component GF will be activated in the absence of CDH1 and in the presence of CDH2 or GF itself." In other words, CDH2 is required transiently for GF activation, in the absence of CDH1.

To support the development and analysis of logical models, a series of software tools have been proposed, often with specific assets (Naldi et al., 2009; Klarner et al., 2017; Paulevé, 2017; Stoll et al., 2017).

The *CoLoMoTo Interactive Notebook*[1] (Naldi et al., 2018b) relies on Docker[2] and Jupyter[3] technologies to assist on editing and sharing reproducible analysis workflows for logical models. In addition to the distribution of a set of software tools to define and analyse Boolean and multi-valued networks, a unified Python interface for each of the integrated tools is provided, greatly easing the execution and chaining of complementary analyses.

This protocol describes in details the usage of the CoLoMoTo Interactive Notebook to provide a reproducible analysis of a recently published model of the signaling network controlling tumor cell invasion and migration. More specifically, we combine different tools (**Table 1**) to compute the model stable states, perform stochastic simulations, compute (sets of) mutations controlling the reachability of specific stable states, and evaluate their efficiency.

2. MATERIALS AND EQUIPMENT

2.1. Executable and Reproducible Model Analysis

This protocol has been actually edited entirely as a Jupyter notebook before being converted to a LaTeX document for journal-specific editing purposes. The original notebook file is provided as Supplemental Material. It can also be visualized and downloaded for execution in the CoLoMoTo Interactive Notebook at https://nbviewer.jupyter.org/gist/pauleve/a86717b0ae8750440dd589f778db428f/Usecase%20-%20Mutations%20enabling%20tumour%20invasion.ipynb.

The blocks beginning with `In [..]` correspond to Jupyter *code cells*, which contain the Python instructions to execute. When relevant, the blocks beginning with `Out [..]` display the result of the last instruction of the corresponding code cell.

Provided Docker and Python are installed, the CoLoMoTo Interactive notebook can be installed by typing and executing the following command[4] on GNU/Linux, macOS, and Microsoft Windows:

```
pip install -U colomoto-docker
```

Once installed, the notebook can be executed by typing

```
colomoto-docker -V 2018-05-29
```

The execution of this command will open a web page with the Jupyter notebook interface, enabling the loading and execution of the code. Note that "SHIFT+ENTER" must be used to execute each code cell. More information on `colomoto-docker` usage can be obtained by typing `colomoto-docker --help` and by visiting https://github.com/colomoto/colomoto-docker.

2.2. Notebook Preparation

This notebook makes use of the following Python modules:

```
In [1]: import ginsim
        import biolqm
        import maboss
        import pypint
        from colomoto_jupyter import tabulate
          # for fixpoint table display
        from itertools import combinations
          # for iterating over sets
        import matplotlib.pyplot as plt
          # for modifying plots
```

3. STEPWISE PROCEDURES

3.1. Model

We analyse a Boolean model of the signaling network controlling cell tumor invasion, which was recently reported in Cohen et al. (2015). This model can be loaded directly from the GINsim model repository at http://ginsim.org/models_repository.

We first show how to use GINsim (Naldi et al., 2018a) to fetch and parse the GINML file (GINsim graph-based XML format,

TABLE 1 | List of software tools used in this notebook.

Tool	Website	Role in this notebook
GINsim	ginsim.org	Model input and display, conversion to bioLQM and NuSMV
bioLQM	colomoto.org/biolqm	Fixpoint computation, conversion to MaBoSS and Pint
MaBoSS	maboss.curie.fr	Stochastic simulations, assess impact of mutations on propensity of reaching phenotypes
Pint	loicpauleve.name/pint	Formal prediction of mutants
NuSMV	nusmv.fbk.eu	Formal verification of phenotypes reachability and stability

[1] Available at http://colomoto.org/notebook
[2] https://docker.com
[3] https://jupyter.org
[4] You may have to use `pip3` instead of `pip` depending on your configuration.

encapsulated in a zginml archive) and display the regulatory graph of the network. To load the model, we copied the URL of the `.zginml` file from the model repository page at http://ginsim.org/node/191. The file is also available as Supplemental Data (Data Sheet 1).

```
In [2]: lrg = ginsim.load("http://ginsim.org/sites/
default/files/SuppMat_Model_Master_Model.zginml")
```

The regulatory graph (using the graphical setting specified in the model file) can be displayed with the following command:

```
In [3]: ginsim.show(lrg)
```

The resulting graphics is reproduced in **Figure 1**.

In this regulatory graph, the gray boxes denote input and output vertices (nodes). Green arrows and red T arrows respectively denote activatory and inhibitory interactions. A set of rules combining the vertices with the Boolean operators NOT, AND, and OR, which must be consistent with the regulatory graph, then allows the computation of enabled transitions for each network state. These rules have been defined in Cohen et al. (2015) and are specified within the GINsim model.

3.2. Identification of Stable States

First, we compute the complete list of logical stable states (or fixpoints) of the model using the Java library `bioLQM` (Naldi, 2018). We thus need to convert the GINsim model into bioLQM:

```
In [4]: lqm = ginsim.to_biolqm(lrg)
```

At that stage, `lrg` is a Python object representing the model suitable for GINsim, and `lqm` is a Python object representing the equivalent model suitable for bioLQM.

The list of stable states of a bioLQM model is computed as follows:

```
In [5]: fixpoints = biolqm.fixpoints(lqm)
```

Here, `fixpoints` is a Python list of states. A state is encoded as a Python association table (dictionary), which maps each node of the network to a value.

For a nice display of the list of stable states, one can use the `tabulate` function provided in the `colomoto_jupyter` Python library, imported at the beginning of the notebook:

```
In [6]: tabulate(fixpoints)
```

Figure 2 shows the table as displayed in the notebook. The complete table is given in Supplemental Data.

It results that the model has nine stable states, each corresponding to a row in the table, four of which enable apoptosis (rows with value 1 in fourth column "Apoptosis"). Note that the input node `DNAdamage` is also active in each of these four states.

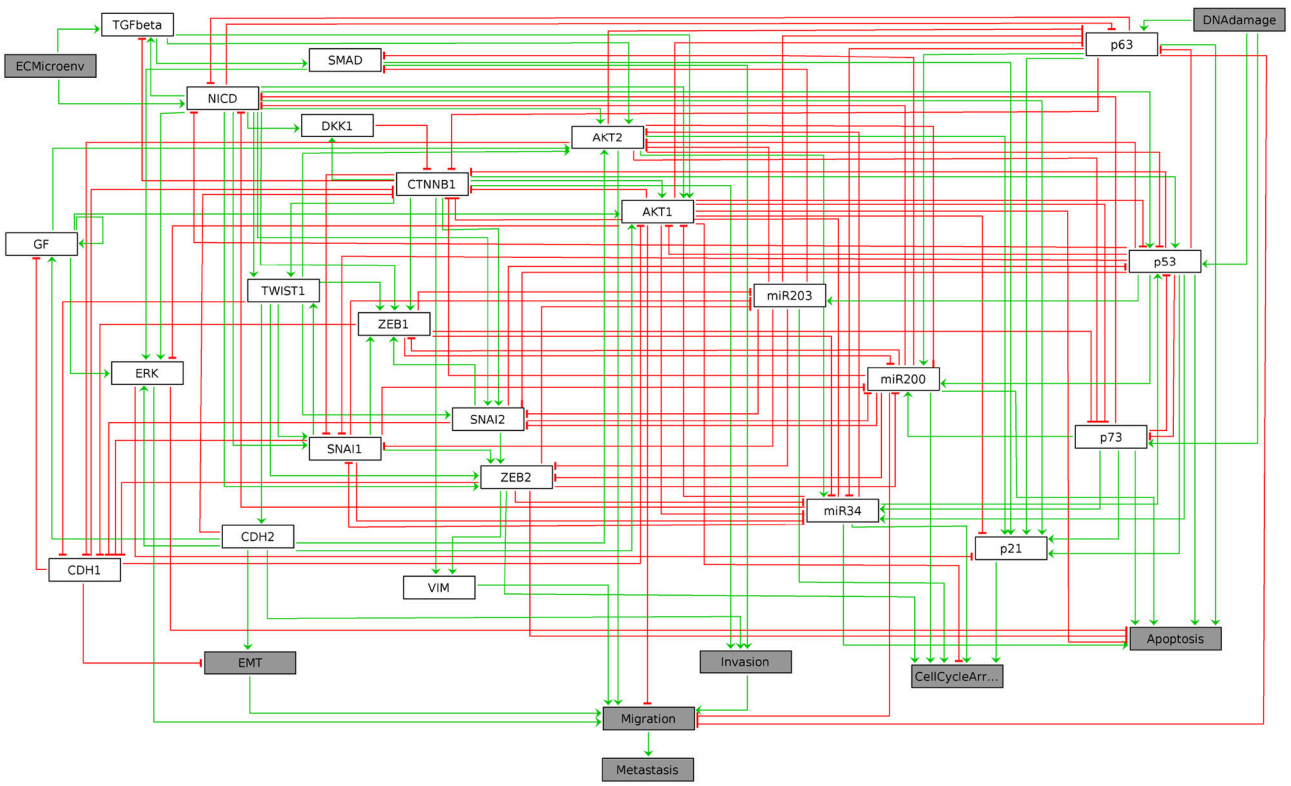


FIGURE 1 | Graphical output resulting from the input code: `In [3]: ginsim.show(lrg)`.

```
In [6]: tabulate(fixpoints)
```

Out[6]:

	ECMicroenv	DNAdamage	Metastasis	Migration	Invasion	EMT	Apoptosis	CellCycleArrest	GF	TGFbeta	p21	CDH1	CDH2	VIM	TWIST1	SNAI1	SNAI
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
1	0	0	0	0	0	1	0	1	1	0	0	0	1	1	1	1	
2	0	1	0	0	0	0	1	1	0	0	1	1	0	0	0	0	
3	0	1	0	0	0	0	1	1	0	0	1	1	0	0	0	0	
4	0	1	0	0	0	1	0	1	1	0	0	0	1	1	1	1	
5	1	0	1	1	1	1	0	1	1	1	0	0	1	1	1	1	
6	1	1	0	0	0	0	1	1	0	1	1	1	0	0	0	0	
7	1	1	0	0	0	0	1	1	0	1	1	1	0	0	0	0	
8	1	1	1	1	1	1	0	1	1	1	0	0	1	1	1	1	

FIGURE 2 | Graphical output resulting from the input code: `In [6]: tabulate(fixpoints)`.

A state can be visualized on the regulatory graph using GINsim. For example, the third stable state can be displayed using the following command:

```
In [7]: ginsim.show(lrg, fixpoints[2])
```

The resulting graphics is reproduced in **Figure 3**.

In this graph, the vertices shown in white or orange denote components that are OFF (value 0) or ON (value 1) respectively.

3.3. Assessing the Probabilities to Reach Alternative Attractors Using MaBoSS

MaBoSS (Stoll et al., 2017) is a C++ software enabling the stochastic simulation of Boolean networks by translating them into continuous time Markov processes. Each node activation and inactivation is associated with an *up* and a *down rate*, which specify the propensity of the corresponding transitions. From a given state, the simulation integrates all the possible node updates and derives a probability and a duration for each transition. By default, all transitions are assigned the same rate. For a given set of initial conditions, MaBoSS produces time trajectories and estimates probabilities of model states over the whole simulation time. Steady state distributions can thus be approximated, provided that a sufficient number of sufficiently long simulations have been performed.

The aim of this section is to reproduce part of the results obtained by Cohen et al. (2015), which show that a Notch (NICD) gain-of-function together with a p53 loss-of-function prevent reaching a stable apoptotic phenotype.

First, we convert the bioLQM model to MaBoSS:

```
In [8]: wt_sim = biolqm.to_maboss(lqm)
```

The variable `wt_sim` is a Python object that gathers both the Boolean network rules and the settings for the simulations, including the transition rates.

3.3.1. Simulation Setup

The stochastic simulation of Boolean networks with MaBoSS requires the specification of several parameters.

3.3.1.1. Initial states

First, a distribution of initial states must be specified: each simulation then starts from a state sampled from this distribution. The distribution is determined by assigning a probability to start in state `0` or in state `1` to each node. By default, a node has a probability `1` to start in state `0`.

The `maboss` Python library provides *widgets* to ease the assignment of this initial distribution. The following code enables the definition of a distribution of initial states with all nodes at `0`, except `DNAdamage` and `ECMicroenv` with equiprobable `0` and `1` values. After pressing "OK," the notebook cell will be replaced by the actual Python call resulting in equal probabilities for these two nodes to start in active or inactive states.

```
In [9]: maboss.wg_set_istate(wt_sim)
```

The notebook will then display the widgets reproduced in **Figure 4**. The selection of nodes and of initial conditions shown in this figure are then translated in the following code:

```
In [9]: #maboss.wg_set_istate(wt_sim)
        maboss.set_nodes_istate(wt_sim,["DNAdamage",
        "ECMicroenv"],[0.5, 0.5])
```

3.3.1.2. Output nodes

Using MaBoSS, we can focus on the *output* nodes and ignore the other nodes, which enable us to identify the corresponding phenotypes. This can be done using the following code:

```
In [10]: #maboss.wg_set_output(wt_sim)
         wt_sim.network.set_output(('Metastasis',
         'Migration', 'Invasion','EMT', 'Apoptosis',
         'CellCycleArrest'))
```

3.3.1.3. Simulation parameters

The `update_parameters` method can be used to specify several parameters for the stochastic simulation algorithm. We show below the complete list of parameters with the values obtained by default when translating a model from GINsim. The method can be called with any subset of these parameters.

Among the parameter list, `sample_count` corresponds to the number of simulations performed to compute statistics, while

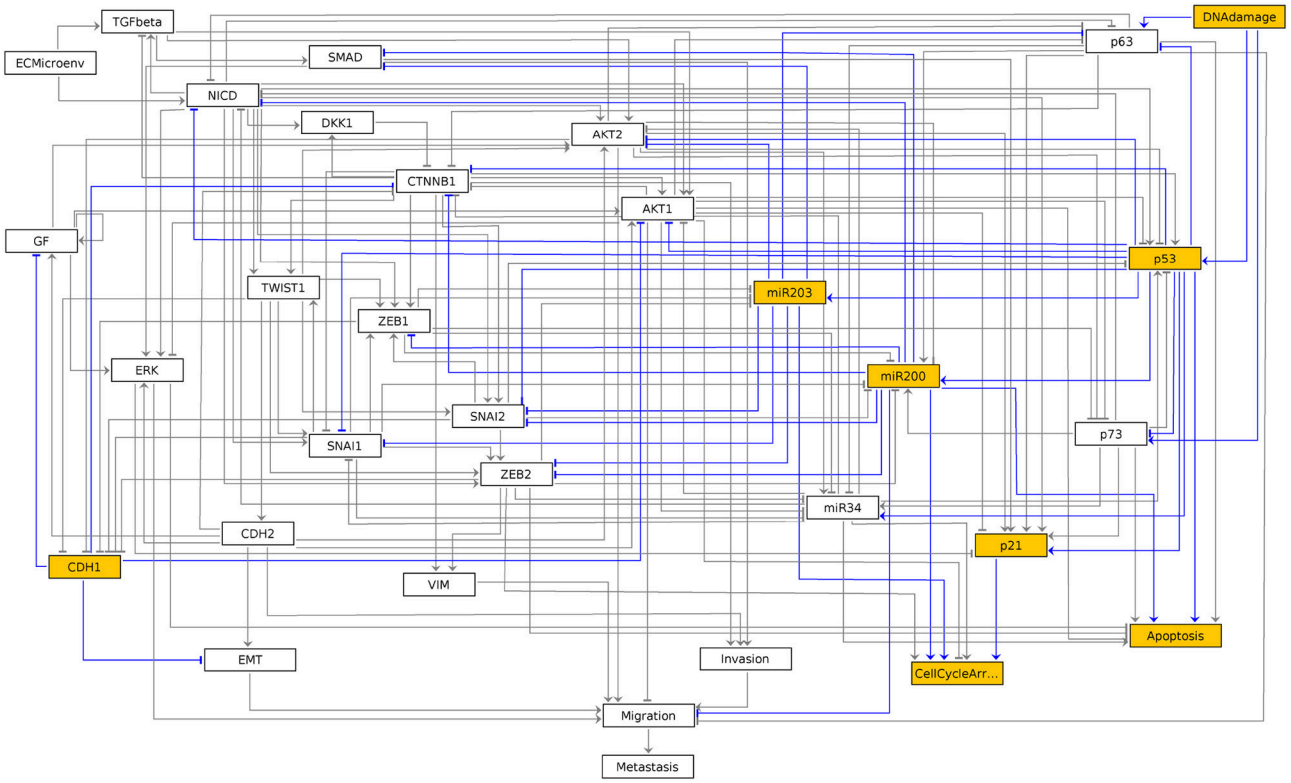


FIGURE 3 | Graphical output resulting from the input code: `In [7]: ginsim.show(lrg, fixpoints[2])`.

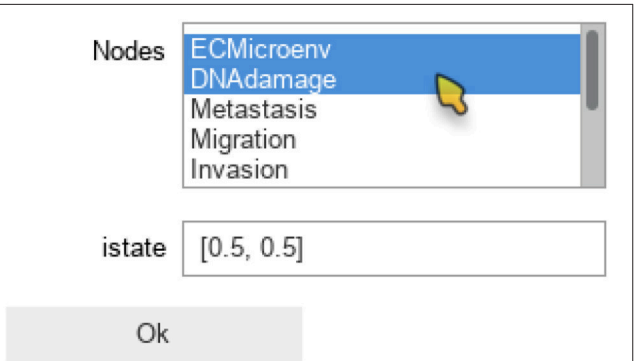


FIGURE 4 | Graphical output resulting from the input code: `In [9]: maboss.wg_set_istate(wt_sim)`.

`max_time` is the maximum (simulated) duration of a trajectory. Note that for a proper estimation of probabilities of the stable states, `max_time` needs to be long enough for the simulation to reach an asymptotic solution.

```
In [11]: wt_sim.update_parameters(discrete_time=0,
         use_physrandgen=0,  seed_pseudorandom=100,
         sample_count=50000,  max_time=75,
         time_tick=0.1,  thread_count=4,
         statdist_traj_count=100,
         statdist_cluster_threshold=0.9)
```

3.3.2. Simulation of the Wild-Type Model

The object `wt_sim` represents the input of MaBoSS, encompassing both the network and simulation parameters. The simulations are triggered with the `.run()` method and return a Python object for accessing the results.

```
In [12]: %time wt_results = wt_sim.run()

CPU times: user 4.61 ms, sys: 406 s,
total: 5.02 ms Wall time: 2.89 s
```

The resulting object gives access to the output data generated by MaBoSS. It includes notably the mean probability over time for the activity of the output states integrated over all the performed simulations.

The function `plot_piechart` displays proportionally the mean probability of each output state at the *last* time point. Provided the simulation time has been set high enough, this gives an approximation of the probabilities of the stable states reachable from the specified initial conditions.

```
In [13]: wt_results.plot_piechart()
```

The resulting graphics is reproduced in **Figure 5**.
In this chart, a state is described by the set of its active output nodes and is associated to a phenotype. For instance, the "`<nil>`" phenotype has all output nodes set

to 0, which was referred to as the "homeostatic state" in the original article; in the case of the "`Apoptosis -- CellCycleArrest`" phenotype, the two output nodes `Apoptosis` and `CellCycleArrest` are simultaneously active, while the other output nodes are inactive; the "`EMT -- CellCycleArrest`" phenotype denotes cells that have gone through the epithelial to mesenchymal transition (EMT), but did not invade the tissue, hence the output nodes `Invasion`, `Migration` and `Metastasis` are inactive; finally the "`Migration -- Metastasis -- Invasion -- EMT -- CellCycleArrest`" phenotype corresponds to a metastatic state, i.e., to cells that went through EMT, invaded the tissue and migrated to a distant site.

From this plot, we can deduce that, from the specified set of initial conditions, the apoptotic state (orange section), the EMT (purple section) and the metastatic states (green section) can be reached (the proportion of simulations that reached none of these phenotypes correspond to the red section).

The mean value of each output node during the simulations can be plotted with the following command:

```
In [14]: wt_results.plot_node_trajectory(until=40)
```

The resulting graphics is reproduced in **Figure 6**.

3.3.3. Simulation of Double Mutant Notch++/p53--

In the original article (Cohen et al., 2015), the authors analyzed the double Notch++/p53-- mutant, i.e., the combination of a Notch gain-of-function combined with a p53 loss-of-function, showing that all trajectories lead to a metastatic state.

A mutant can be configured by copying the wild-type model, and use the `mutate` method to model the desired gains and losses of function:

```
In [15]: mut_sim = wt_sim.copy()
         mut_sim.mutate("p53", "OFF")
         mut_sim.mutate("NICD", "ON")
```

The modified model can then be simulated exactly as for the wild-type case:

```
In [16]: %time mut_results = mut_sim.run()
```

```
CPU times: user 5.13 ms, sys: 137 s,
total: 5.27 ms Wall time: 2.99 s
```

```
In [17]: mut_results.plot_piechart()
```

The resulting graphics is reproduced in **Figure 7**.

Using the same parameters as for the wild-type model, all the trajectories obtained for the double mutant model reach the metastatic invasive state exclusively. This suggests that such a double mutation can be responsible for a loss of apoptotic capability of cancer cells.

3.4. Formal Analysis With Pint and NuSMV

In the above section, the conclusion regarding the loss of apoptotic stable state relies on stochastic simulations, which, in general, may not offer a complete coverage of the possible trajectories. Therefore, one may want to formally verify whether the loss of reachable stable apoptosis state is total or not. First, we show how to use Pint (Paulevé, 2017) to predict combinations of mutations which are guaranteed to prevent the activation of apoptosis. Next, we use the software NuSMV (Cimatti et al., 2002) to evaluate formally the Notch++/p53--

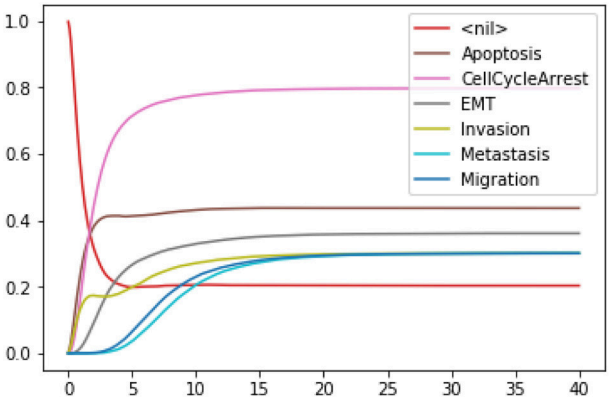


FIGURE 6 | Graphical output resulting from the input code: `In [14]: wt_results.plot_node_trajectory(until=40)`.

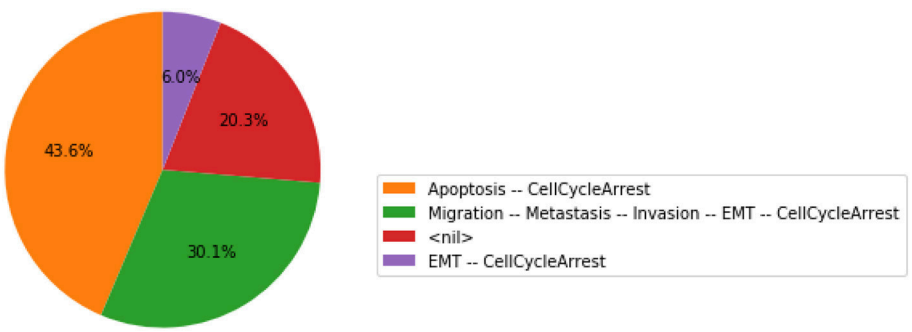


FIGURE 5 | Graphical output resulting from the input code: `In [13]: wt_results.plot_piechart()`.

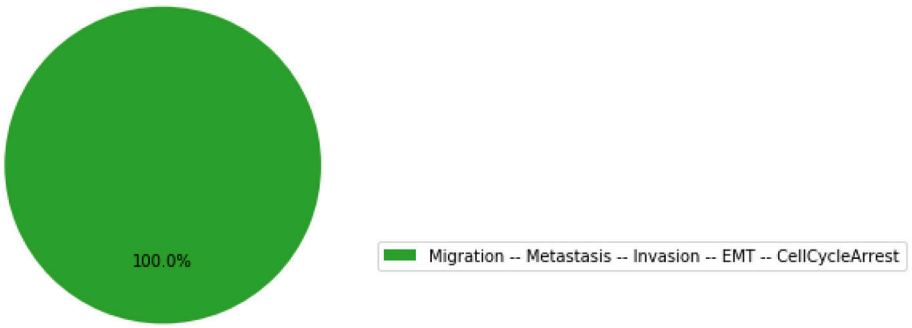


FIGURE 7 | Graphical output resulting from the input code: `In [17]: mut_results.plot_piechart()`.

double mutant. Finally, we use MaBoSS to assess the efficiency of new combinations of mutations predicted by Pint.

3.4.1. Formal Predictions of Mutations From the Wild-Type Model

Pint implements formal methods that allow deducing combinations of mutations guaranteed to block the reachability of a given state.

First, we convert the bioLQM model to Pint:

```
In [18]: an = biolqm.to_pint(lqm)
```

Then, we transfer the initial conditions defined in MaBoSS to the Pint model `an`. Like MaBoSS, Pint supports multiple initial values for a single node. However, in contrast to MaBoSS, Pint does not consider probability distributions.

```
In [19]: an.initial_state.update(wt_sim.get_initial_
         state())
         an.initial_state.changes()
         # display non-default (0) initial value

Out[19]: {'DNAdamage': (0, 1), 'ECMicroenv': (0, 1)}
```

Given a (partial) state specification, Pint provides the method `oneshot_mutations_for_cut`, which returns different sets of mutations guaranteed to prevent any trajectory from any possible initial state to reach, *even transiently*, the specified state.

```
In [20]: %time  \
      an.oneshot_mutations_for_cut(Apoptosis=1,  \
         exclude={"ECMicroenv",  "DNAdamage"})

CPU times: user 6.11 ms, sys: 158 s,
total: 6.27 ms Wall time: 191 ms


Out[20]: [{'ZEB2': 1},
          {'AKT1': 1},
          {'AKT2': 1},
          {'ERK': 1},
          {'NICD': 1, 'SNAI2': 1, 'ZEB1': 1},
          {'SNAI2': 1, 'ZEB1': 1, 'p63': 0},
          {'SNAI2': 1, 'ZEB1': 1, 'miR203': 1},
          {'NICD': 1, 'SNAI2': 1, 'p73': 0},
          {'SNAI2': 1, 'p63': 0, 'p73': 0},
          {'SNAI2': 1, 'miR203': 1, 'p73': 0},
          {'NICD': 1, 'ZEB1': 1, 'p53': 0},
          {'ZEB1': 1, 'p53': 0, 'p63': 0},
          {'ZEB1': 1, 'miR203': 1, 'p53': 0},
          {'NICD': 1, 'p53': 0, 'p73': 0},
          {'p53': 0, 'p63': 0, 'p73': 0},
          {'miR203': 1, 'p53': 0, 'p73': 0}]
```

Among the returned mutation sets, one can spot the mutation `{'NICD': 1, 'p53': 0, 'p73': 0}`, which combines a gain-of-function of Notch (`'NICD': 1`) with a loss-of-function of p53 (`'p53': 0`), along with a loss-of-function of p73 (`'p73': 0`).

Noteworthy, forbidding *transient* reachability entails a stronger constraint than just preventing any *stable* state with the specified property. Indeed, some mutations may remove the stability of the specified states, while some trajectories may still traverse these states, but only transiently.

Therefore, the sets of mutations returned by Pint, albeit correct, might be non-minimal for controlling only the long-term dynamics of the system. Finally, note that the analysis of Pint can give incomplete results. This is due to the technology on which the computation relies (static analysis), which allows addressing very large scale networks.

3.4.2. Revisiting the Notch++/p53-- Double Mutant

We will first formally analyse the Notch++/p53-- double mutant to show that asymptotic apoptosis is forbidden, although transient activation of apoptosis node might still be possible.

One can apply a mutation on a Pint model using the `lock` method. A new model is returned with a constant value for the corresponding nodes.

```
In [21]: mut_an = an.lock(NICD=1, p53=0)
```

Then, we use the temporal logic CTL (Clarke and Emerson, 1982) to specify formally the dynamical properties to verify. CTL expression can be built using the `colomoto.temporal_logics` Python module.

```
In [22]: from colomoto.temporal_logics import *
```

First, the existence of a trajectory leading to a *transient* state where `Apoptosis` is active can be specified as follows:

```
In [23]: transient_apoptosis = EF(S(Apoptosis=1))
```

`EF` is a temporal logic operator that is true if there exists at least one trajectory leading to a state verifying the properties given as argument. Here the property `S(Apoptosis=1)` specifies that the state has the node `Apoptosis` active.

Next, the existence of a trajectory leading to a *stable* `Apoptosis` activation can be specified as follows:

```
In [24]: stable_apoptosis = EF(AG(S(Apoptosis=1)))
```

Here, `AG` enforces that *all* the states reachable via any trajectory have the node `Apoptosis` active.

Finally, we gather these two properties in a Python dictionary for later use:

```
In [25]: ctl_specs = {
             "reach-apoptosis": transient_apoptosis,
             "stable-apoptosis": stable_apoptosis
         }
```

The adequation of a model with a CTL property can be assessed using a *model-checker* such as NuSMV (Abou-Jaoudé et al., 2015).

Pint provides a conversion to NuSMV models. By default, the NuSMV model considers any initial state. With the `skip_init=False` option, we enforce that the properties are verified only from the initial states defined earlier.

```
In [26]: smv = mut_an.to_nusmv(skip_init=False)
```

We then add the properties defined above, and ask NuSMV to verify them.

```
In [27]: smv.add_ctls(ctl_specs)
         %time smv.verify()

CPU times: user 0 ns, sys: 4.68 ms,
total: 4.68 ms Wall time: 12.4 s


Out[27]: {'reach-apoptosis': True,
          'stable-apoptosis': False}
```

Interestingly, the Notch++/p53-- double mutant can still reach an apoptotic state, but only transiently: the property `stable-apoptosis` being false, it is guaranteed that all trajectories eventually lead to stable apoptosis inactivation.

To complete our analysis, we now consider the triple mutant obtained by adding a loss-of-function of `p73`. As predicted by Pint, transient reachability of apoptosis is impossible in this triple mutant. We can use NuSMV to further verify that it is the case, using the following code:

```
In [28]: smv_mut3 = an.lock(NICD=1, p53=0,  \
             p73=0) .to_nusmv(skip_init=False)
         smv_mut3.add_ctls(ctl_specs)
         smv_mut3.verify()

Out[28]: {'reach-apoptosis': False,
          'stable-apoptosis': False}
```

3.4.3. Analysis of Formally Predicted SNAI2++/ZEB1++/miR203++ Triple Mutant

The mutant combinations predicted with Pint should be refined when the aim is to control specifically stable behaviors. In general, given a set of mutations guaranteed to block any transient activation of a node, one may verify whether only a subset of them are sufficient to achieve proper control of the sole stable states.

We show here how we can take advantage of the Python environment to provide a small program, which, for each subset of mutations of a multiple mutant (here a triple gain-of-function for `SNAI2`, `ZEB1` and `miR203`), performs stochastic simulations with MaBoSS to assess the probabilities to reach the different stable behaviors from the specified set of states.

The computation can take a couple of minutes. The results are shown in a graphical form (colored pie charts) for each single and double loss-of-function combination. In the pie charts, "Others" regroup states with an individual probability less than 1%, which often correspond to simulated trajectories having not reached an attractor in the given amount of time.

```
In [29]:
formal_mutant = {'SNAI2': 1, 'ZEB1': 1, 'miR203': 1}
for i in [1, 2]:
   # for any subset of mutations of size 1 then 2
   for mutants in combinations(formal_mutant, i):
     # copy the wild-type MaBoSS model
     masim = wt_sim.copy()
     # apply the mutations
     for m in mutants:
        masim.mutate(m, "ON" if formal _mutant[m] \
                                 else "OFF")
     # run the simulations
     mares = masim.run()
     # plot the piechart of stable states
     mares.plot_piechart()
     # print the mutation in the title
     def mutname(m):
        return m + ("++" if formal_mutant[m]  \
                                 else "--")
     name = "/".join(map(mutname,mutants))
     plt.title("%s mutant" % name)
```

The resulting graphics are reproduced in **Figures 8–13**.

Note that only one of the pie charts shows an absence of apoptotic state: the SNAI2++/miR203++ double mutant (**Figure 13**).

This can be formally verified with NuSMV, as we did for the Notch++/p53-- mutant:

```
In [30]: smv_mut_test = an.lock(SNAI2=1, \
             miR203=1).to_nusmv(skip_init=False)
         smv_mut_test.add_ctls(ctl_specs)
         smv_mut_test.verify()

Out[30]: {'reach-apoptosis': True,
          'stable-apoptosis': False}
```

4. ANTICIPATED RESULTS

With this protocol, we showed how the Python interface and Jupyter integration of GINsim, bioLQM, MaBoSS, and Pint ease

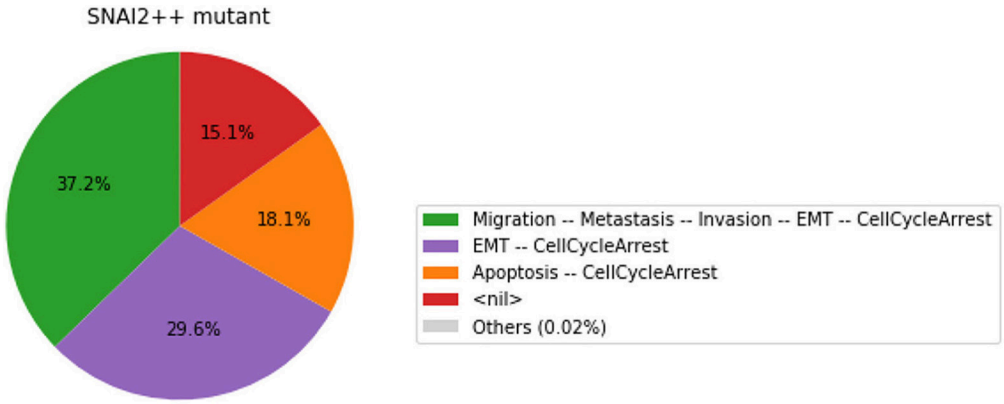


FIGURE 8 | Graphical output resulting from the input code: `In [29]`.

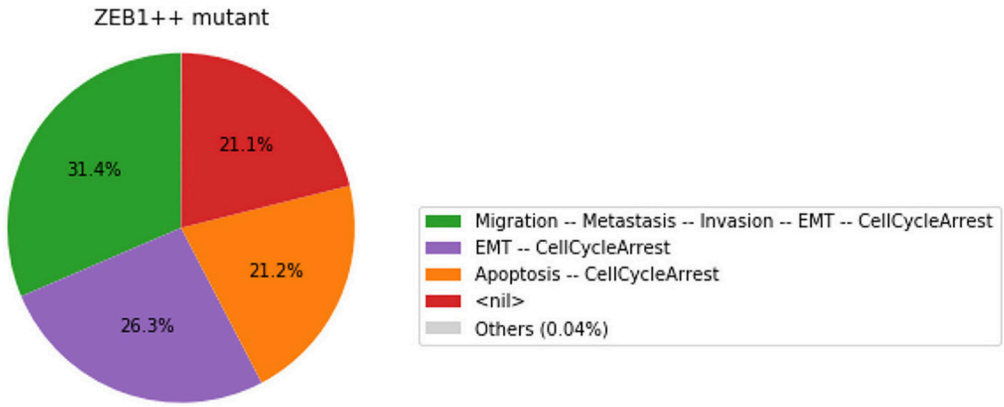


FIGURE 9 | Graphical output resulting from the input code: `In [29]`.

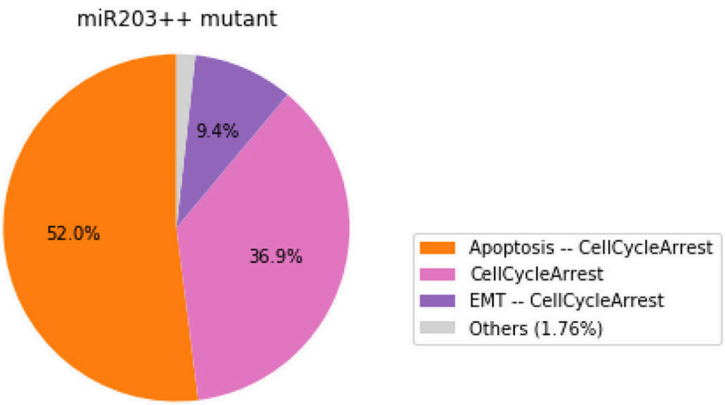


FIGURE 10 | Graphical output resulting from the input code: `In [29]`.

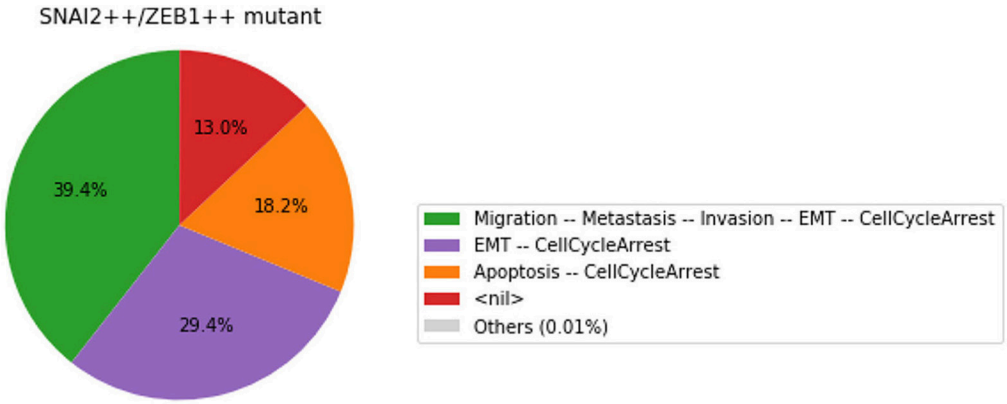


FIGURE 11 | Graphical output resulting from the input code: `In [29]`.

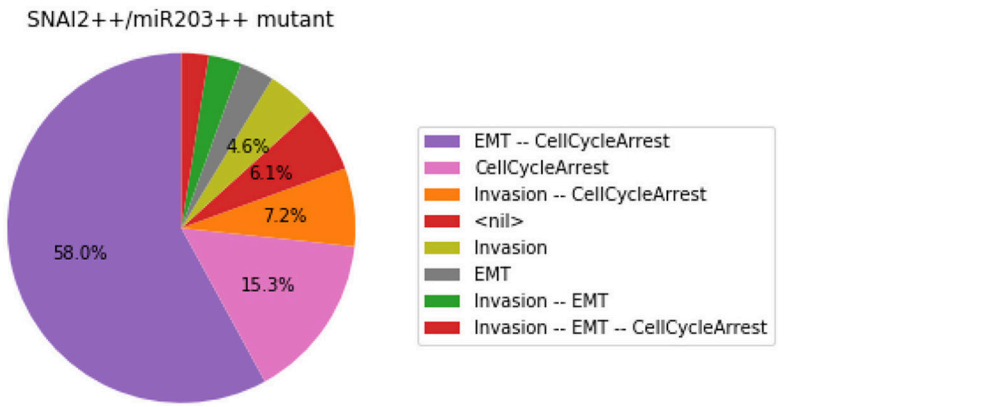


FIGURE 12 | Graphical output resulting from the input code: `In [29]`.

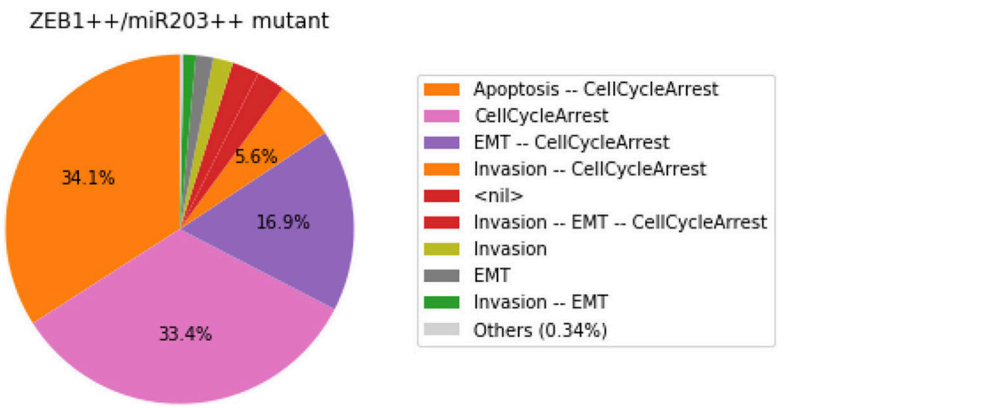


FIGURE 13 | Graphical output resulting from the input code: `In [29]`.

the delineation of sophisticated re-executable computational analyses of qualitative models of biological networks, combining and chaining different software with a unified interface.

Leaning on the CoLoMoTo Docker image and on the companion Jupyter notebook, we have demonstrated the benefits of this framework by revisiting the analysis of a recent Boolean model of the signaling network controlling cancer cell metastasis. We could reproduce results previously obtained with GINsim and MaBoSS, which demonstrate that the Notch++/p53-- double mutant can suppress the apoptotic outcome. Furthermore, a formal analysis of trajectories with Pint enabled us to deduce novel "anti-apoptotic" combinations of mutations, including a triple mutant that forbids even transient activation of apoptosis, which were subsequently quantified using MaBoSS.

The predicted of mutations point to potential synergistic genetic interactions underlying uncontrolled tumor proliferation. These combinations would deserve further analysis, in particular regarding potential correlations with specific clinical outcomes. For example, one could check whether the loss of apoptosis triggering correlates with higher tumor grades.

Similar computational analyses could be performed to predict combinations of perturbations enforcing the existence of a given stable phenotype, e.g., apoptosis, which could then serve as a basis to design novel therapeutic strategies.

AUTHOR CONTRIBUTIONS

NL, AN, CH, LP implemented the necessary Python modules, their integration in the Jupyter interface, and the Docker image. NL, AN, GS, DT, AZ, LC, LP participated to the general design of the notebook. All authors participated to the writing of the article.

FUNDING

DT and CH acknowledge support from the French Plan Cancer, in the context of the projects CoMET (2014-2017) and SYSTAIM (2015-2019). DT and AN acknowledge support from the French Agence Nationale pour la Recherche (ANR), in the context of the project SCAPIN [ANR-15-CE15-0006-01]. AZ acknowledges support by the Ministry of education and science of Russia (Project No. 14.Y26.31.0022). AZ and LC acknowledge support from ITMO Cancer, in the context of the INVADE grant (Call Systems Biology 2012), and from the EU ERACoSysMed programme, in the context of the COLOSYS project. AZ, LC, and LP acknowledge support from the ANR in context of the ANR-FNR project AlgoReCell [ANR-16-CE12-0034]. LP acknowledge support from Paris Ile-de-France Region (DIM RFSI) and Labex DigiCosme [ANR-11-LABEX-0045-DIGICOSME] operated by ANR as part of the program Investissement d'Avenir Idex Paris-Saclay [ANR-11-IDEX-0003-02].

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphys.2018.00787/full#supplementary-material

The supplemental data "Notebook" (Data Sheet 1) includes the source notebook file (`.ipynb` extension) which can be uploaded and executed within the Jupyter interface of the CoLoMoTo notebook, using the Docker image `colomoto/colomoto-docker:2018-05-29`. We further provide a static HTML file to preview the Jupyter rendering of the notebook, along with the file containing the Boolean model used.

The supplemental data "Fixpoints" (Data Sheet 2) gives the complete description of the fixpoints computed by code cell `In [6]`.

REFERENCES

Abou-Jaoudé, W., Monteiro, P. T., Naldi, A., Grandclaudon, M., Soumelis, V., Chaouiya, C., et al. (2015). Model checking to assess t-helper cell plasticity. *Front. Bioeng. Biotechnol.* 2:86. doi: 10.3389/fbioe.2014.00086

Cimatti, A., Clarke, E., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., et al. (2002). "NuSMV 2: An opensource tool for symbolic model checking," in *Computer Aided Verification, Vol. 2404 of Lecture Notes in Computer Science*, eds E. Brinksma and K. G. Larsen (Copenhagen: Springer), 359–364. doi: 10.1007/3-540-45657-0_29

Clarke, E. M., and Emerson, E. A. (1982). "Design and synthesis of synchronization skeletons using branching-time temporal logic," in *Logic of Programs*, ed D. Kozen (New York, NY: Springer), 52–71. doi: 10.1007/BFb0025774

Cohen, D. P. A., Martignetti, L., Robine, S., Barillot, E., Zinovyev, A., and Calzone, L. (2015). Mathematical modelling of molecular pathways enabling tumour cell invasion and migration. *PLoS Comput. Biol.* 11:e1004571. doi: 10.1371/journal.pcbi.1004571

Collombet, S., van Oevelen, C., Sardina Ortega, J. L., Abou-Jaoudé, W., Di Stefano, B., Thomas-Chollier, M., et al. (2017). Logical modeling of lymphoid and myeloid cell specification and transdifferentiation. *Proc. Natl. Acad. Sci. U.S.A.* 114, 5792–5799. doi: 10.1073/pnas.1610622114

Helikar, T., Kowal, B., McClenathan, S., Bruckner, M., Rowley, T., Madrahimov, A., et al. (2012). The cell collective: toward an open and collaborative approach to systems biology. *BMC Syst. Biol.* 6:96. doi: 10.1186/1752-0509-6-96

Klarner, H., Streck, A., and Siebert, H. (2017). Pyboolnet: a python package for the generation, analysis and visualization of boolean networks. *Bioinformatics* 33, 770–772. doi: 10.1093/bioinformatics/btw682

Naldi, A. (2018). bioLQM: a java library for the manipulation and conversion of Logical Qualitative Models of biological networks. *bioRxiv*. doi: 10.1101/287011

Naldi, A., Berenguier, D., Fauré, A., Lopez, F., Thieffry, D., and Chaouiya, C. (2009). Logical modelling of regulatory networks with GINsim 2.3. *Biosystems* 97, 134–139. doi: 10.1016/j.biosystems.2009.04.008

Naldi, A., Hernandez, C., Abou-Jaoudé, W., Monteiro, P. T., Chaouiya, C., and Thieffry, D. (2018a). Logical modelling and analysis of cellular regulatory networks with GINsim 3.0. *Front. Physiol.* 9:646. doi: 10.3389/fphys.2018.00646

Naldi, A., Hernandez, C., Levy, N., Stoll, G., Monteiro, P. T., Chaouiya, C., et al. (2018b). The CoLoMoTo interactive notebook: accessible and reproducible computational analyses for qualitative biological networks. *Front. Physiol.* 9:680. doi: 10.3389/fphys.2018.00680

Paulevé, L. (2017). "Pint: a static analyzer for transient dynamics of qualitative networks with IPython interface," in *CMSB 2017 - 15th Conference on*

Computational Methods for Systems Biology, Vol. 10545 of Lecture Notes in Computer Science, eds J. Feret and H. Koeppl (Darmstadt: Springer), 309–316. doi: 10.1007/978-3-319-67471-1_20

Stoll, G., Caron, B., Viara, E., Dugourd, A., Zinovyev, A., Naldi, A., et al. (2017). MaBoSS 2.0: an environment for stochastic Boolean modeling. *Bioinformatics* 33, 2226–2228. doi: 10.1093/bioinformatics/btx123

Zaudo, J. G. T., and Albert, R. (2015). Cell fate reprogramming by control of intracellular network dynamics. *PLoS Comput. Biol.* 11:e1004193. doi: 10.1371/journal.pcbi.1004193

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

ORIGINAL RESEARCH
published: 17 July 2018
doi: 10.3389/fphys.2018.00774

Global Stabilization of Boolean Networks to Control the Heterogeneity of Cellular Responses

Jung-Min Yang[1], *Chun-Kyung Lee*[1] *and Kwang-Hyun Cho*[2]*

[1] School of Electronics Engineering, Kyungpook National University, Daegu, South Korea, [2] Laboratory for Systems Biology and Bio-inspired Engineering, Department of Bio and Brain Engineering, Korea Advanced Institute of Science and Technology, Daejeon, South Korea

Boolean networks (BNs) have been widely used as a useful model for molecular regulatory networks in systems biology. In the state space of BNs, attractors represent particular cell phenotypes. For targeted therapy of cancer, there is a pressing need to control the heterogeneity of cellular responses to the targeted drug by reducing the number of attractors associated with the ill phenotypes of cancer cells. Here, we present a novel control scheme for global stabilization of BNs to a unique fixed point. Using a sufficient condition of global stabilization with respect to the adjacency matrix, we can determine a set of constant controls so that the controlled BN is steered toward an unspecified fixed point which can then be further transformed to a desired attractor by subsequent control. Our method is efficient in that it has polynomial complexity with respect to the number of state variables, while having exponential complexity with respect to in-degree of BNs. To demonstrate the applicability of the proposed control scheme, we conduct simulation studies using a regulation influence network describing the metastatic process of cells and the Mitogen-activated protein kinase (MAPK) signaling network that is crucial in cancer cell fate determination.

Keywords: Boolean networks (BNs), global stabilization, sequential control, heterogeneity, systems biology

OPEN ACCESS

Edited by:
Matteo Barberis,
University of Amsterdam, Netherlands

Reviewed by:
Reka Albert,
Pennsylvania State University, United States
Loïc Paulevé,
Délégation Ile-de-France Sud (CNRS), France

***Correspondence:**
Kwang-Hyun Cho
ckh@kaist.ac.kr

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: *31 January 2018*
Accepted: *04 June 2018*
Published: *17 July 2018*

Citation:
Yang J-M, Lee C-K and Cho K-H (2018) Global Stabilization of Boolean Networks to Control the Heterogeneity of Cellular Responses. Front. Physiol. 9:774. doi: 10.3389/fphys.2018.00774

1. INTRODUCTION

As a biology-based interdisciplinary field, systems biology is receiving a great interest in recent years as it can investigate complex interactions within biological systems using holistic approaches to biological research (Park et al., 2006; Kim et al., 2007; Murray et al., 2010). Since first proposed by Kauffman (1969), Boolean networks (BNs) have been successfully applied to modeling gene regulatory networks in systems biology. The main reason for utilizing BNs is that they can formulate simplified dynamics of biological networks while capturing the essential characteristics of the networks. Since each gene in the network can be considered to have approximately two levels of activity—active (logical one) or inactive (logical zero), one can define the corresponding Boolean state variables and Boolean logics that serve as state transition functions.

Attractors are the most important factor of BNs as they represent key cellular phenotypes. It is known that finding singleton attractors, or *fixed points* as they are often called, is an NP-hard problem (Akutsu et al., 1998). Nevertheless, many studies exist in the literature on detecting and analyzing attractors in the framework of BNs; see, e.g., Helikar and Rogers (2009); Cheng et al. (2011a); Gonzalez et al. (2006); Zheng et al. (2013); Cheng et al. (2017); Zheng et al. (2016) and references therein.

On the other hand, controlling a cellular behavior is becoming an important issue in systems biology (Liu et al., 2011; Cornelius et al., 2013; Wang et al., 2016). In particular, inducing homogeneous cellular responses is critical to deal with tumor heterogeneity in most of the anti-cancer therapies (Burrell et al., 2013; Mroz et al., 2015; McGranahan and Swanton, 2017). Recent studies confirm that non-genetic heterogeneity is the key driving force for the evolution of cancerous cells (Brock et al., 2009; Shaffer et al., 2017; Dagogo-Jack and Shaw, 2018). In terms of attractors, this is a problem of controlling BNs so that the controlled BN can always converge to one or a smaller number of attractors among all possible ones. It is most desirable if we can reduce the number of undesired attractors selectively. If not, as the second best policy, we can consider the two-step strategy where we first drive the BN toward a global attractor and then transform it into a desired one in the second step. In this way, the desired attractor landscape can have one fixed point.

In this paper, we address the aforementioned problem, termed global stabilization of BNs. The main objective is to determine a set of constant controls that drive the BN toward a unique fixed point. There are many recent results on global stabilization of BNs. Notable among them is Cheng et al. (2011b) that presents necessary and sufficient conditions for global stability of BNs based on a matrix operation called semi-tensor product (STP). In Kim et al. (2013), on the other hand, a minimal set of state variables that make the BN reach a desired attractor is defined as the control kernel, and a general algorithm for the identification of the control kernel is presented. In Zañudo and Albert (2015), attractors are represented by stable motifs and a method is proposed to identify control targets that ensure the convergence of the BN to a desired attractor. The approach in Zañudo and Albert (2015) is remarkable since it combines the structural and functional information of the BN in finding control targets. In Zañudo et al. (2017), a scheme of feedback vertex set control is proposed that drives biological systems described by general non-linear dynamics (including BNs) toward a desired attractor. Recently, Biane and Delaplace (2017) proposed an elegant theoretical scheme that can stabilize a BN in which abduction-based inference is employed to determine constant control inputs using integer linear programming (ILP). While their method guarantees global stabilization, it needs exponential complexity in deriving control targets. Further, if the dynamics of the BN alters by mutations, ILP must be re-formulated. On the other hand, as will be shown later, our method can be applied to BNs having mutations that cause constitutive activity or inactivity of proteins without any modification from the problem setting of a normal case.

In the present study, we adopt the result of Robert (1986) and Cheng et al. (2011b) to determine constant controls that ensure global stability of BNs. In particular, we utilize the sufficient condition that if the influence graph of a BN is acyclic, there is only one fixed point and from each state there should be a trajectory to it. Our method takes a general BN and will search for a set of control inputs so that the resultant influence graph becomes acyclic. Also, the selection of control inputs relies on the canalization effect of a state variable. A canalized state transition function is fixed to a constant when one state variable belonging to the function as an argument is fixed (Kauffman et al., 2004). As the number of state transition functions canalized by a chosen control input increases, the tendency of the controlled BN directing toward global stabilization is becoming higher. In this regard, we will use the canalization effect as another criterion for selecting control inputs.

Note that "global stabilization" in this paper does not mean that the controlled BN converges to a unique fixed point for all the possible combinations of external inputs and mutation profiles. Since activation and inhibition of some genes is determined only by external inputs representing extra-cellular micro-environments or mutations occurring to the genes, the global attractor cannot be always the same. Rather, the essence of the proposed methodology is the ability to provide a consistent set of control inputs that can achieve global stabilization for any given combination of external inputs and mutation profiles. Though the global attractor may vary depending on external inputs and mutations, our solution guarantees global stabilization despite the difference.

The rest of this paper is organized as follows. Basic notations and terminologies of BNs and relevant notions are introduced in section 2. In section 3, we propose an algorithm for determining a set of constant control inputs that make the BN converge to a unique fixed point. Permutation of the adjacency matrix and canalization by state variables are incorporated into an efficient procedure of determining control inputs. To demonstrate the applicability of the proposed control scheme, numerical experiments are conducted in section 4 where we apply the proposed method to a regulation influence network describing the metastatic process of cells (Cohen et al., 2015) and an MAPK signaling network regulating cancer cell fate determination (Grieco et al., 2013). A comparative study with feedback vertex set control, the control kernel method, and the stable motif control is also provided to highlight the efficiency of the proposed scheme.

2. PRELIMINARIES

$\mathbb{N}$ is the set of natural numbers and $[n]=\{1,\ldots,n\}$ for $n \in \mathbb{N}$. For a finite set A, $|A| \in \mathbb{N}$ denotes the cardinality of A.

A BN with n binary state variables $x_1,\ldots,x_n \in \{0,1\}$ is represented by a Boolean mapping $F = (f_1,\ldots,f_n)^T : \{0,1\}^n \to \{0,1\}^n$ where $f_i : \{0,1\}^n \to \{0,1\}$ is the state transition equation of x_i. Index i and x_i will be used interchangeably for the ith state variable. Letting $x = (x_1,\ldots,x_n)^T$, we express the state evolution with $x := F(x)$. Although our study focuses on BNs with synchronous updating, it can be also applied to asynchronously updating BNs.

The connectivity of F is described by a Boolean matrix with respect to the influence graph of F (Paulevé and Richard, 2012), a topological representation of F in which state variables serve as nodes and there is an edge $x_i \to x_j$ when f_j depends on x_i.

Definition 1. *Given F, the adjacency matrix $A(F)$ is an $n \times n$ Boolean matrix whose (i,j) entry $A_{i,j}(F)$ is 1 if there exists a state $(x_1,\ldots,x_{j-1},0,x_{j+1},\ldots,x_n)^T$ such that*

$f_i(x_1, \ldots, x_{j-1}, 0, x_{j+1}, \ldots, x_n) \neq f_i(x_1, \ldots, x_{j-1}, 1, x_{j+1}, \ldots, x_n)$; *otherwise,* $A_{i,j}(F) = 0$.

$A_{i,j}(F)$ is equal to 1 when x_i is directly affected by x_j. Letting $A(F) = (a_{i,j})$, denote by $a_i^r \in \{0,1\}^{1\times n}$ and $a_j^c \in \{0,1\}^{n\times 1}$ the ith row vector and jth column vector of $(a_{i,j})$, respectively. The norm of each vector is defined as:

$$|a_i^r| = \sum_{j=1}^{n} a_{i,j}$$
$$|a_j^c| = \sum_{i=1}^{n} a_{i,j}$$

$|a_i^r|$ and $|a_j^c|$ are equal to the number of all the incoming and outgoing edges of x_i and x_j, respectively. For the row vector a_i^r and the column vector a_j^c, we define additional parameters:

$$d(a_i^r) = \sum_{j=1}^{i} a_{i,j}$$
$$d(a_j^c) = \sum_{i=1}^{j} a_{i,j}$$

to denote the sum of all one entries from the first to ith position of a_i^r and from the first to jth position of a_j^c, respectively.

Example 1. *Consider the following synthetic BN* $F = (f_1, \ldots, f_8)^T$ *with*

$$\begin{aligned} f_1(x) &= \neg x_3 \wedge x_7 \wedge \neg x_8 \\ f_2(x) &= (x_5 \vee x_6) \wedge \neg x_8 \\ f_3(x) &= x_8 \\ f_4(x) &= x_2 \wedge \neg x_7 \\ f_5(x) &= x_2 \vee x_4 \\ f_6(x) &= x_3 \wedge \neg x_8 \\ f_7(x) &= x_2 \wedge \neg x_8 \\ f_8(x) &= \neg(x_1 \vee x_2) \wedge (x_4 \vee x_7) \end{aligned} \tag{1}$$

where $\neg$, $\wedge$, *and* $\vee$ *are negation, conjunction, and disjunction operation, respectively. In view of Definition 1, the adjacency matrix* $A(F) = (a_{i,j}) \in \{0,1\}^{8\times 8}$ *is derived as*

$$(a_{i,j}) = \begin{pmatrix} 0&0&1&0&0&0&1&1 \\ 0&0&0&0&1&1&0&1 \\ 0&0&0&0&0&0&0&1 \\ 0&1&0&0&0&0&1&0 \\ 0&1&0&1&0&0&0&0 \\ 0&0&1&0&0&0&0&1 \\ 0&1&0&0&0&0&0&1 \\ 1&1&0&1&0&0&1&0 \end{pmatrix}$$

F has three attractors σ_1–σ_3 *where* σ_1 *and* σ_2 *are fixed points and* σ_3 *is a cycle with length 2:*

$$\sigma_1 = (1,1,0,0,1,0,1,0)$$
$$\sigma_2 = (0,0,0,0,0,0,0,0)$$
$$\sigma_3 = (0,1,0,0,1,1,0,1) \leftrightarrows (0,0,1,1,1,0,0,0)$$

Figure 1 *shows the influence graph of* $A(F)$, *where the arrows with pointed heads represent activation and those with bar heads represent inhibition.*

In Cheng et al. (2011b), the necessary and sufficient condition for global stability of F to a fixed point is presented in terms of the adjacency matrix.

Theorem 1. *A BN F globally converges to a unique fixed point if and only if there exists* $k \in \mathbb{N}$ *such that* $A(F^k) = \mathbf{0}_{n\times n}$ *where* F^k *denotes the kth iterate of F.*

$A(F^k) = \mathbf{0}_{n\times n}$ implies that all the edges between state variables are disconnected after the kth iteration. Hence F will reach a unique fixed point for any initial state. However, since the computation of F^k has exponential complexity with respect to n, this criterion is difficult to apply when n is large. Cheng et al. (2011b) also presents a sufficient condition for the existence of a global attractor with polynomial complexity.

Theorem 2. *For a BN F, assume that there exists* $k \in \mathbb{N}$ *such that* $(A(F))^k = \mathbf{0}$, *where* $(A(F))^k$ *denotes Boolean power in which the sum and product operations in the matrix multiplication are logical OR and AND, respectively. Then, F globally converges to a unique fixed point.*

The existence of $k \in \mathbb{N}$ leading to $(A(F))^k = \mathbf{0}$ can be determined by checking whether $A(F)$ falls under a specific category as stated below (Robert, 1986).

Theorem 3. *For a BN F, there exists* $k \in \mathbb{N}$ *such that* $(A(F))^k = \mathbf{0}$ *if and only if a permutation matrix* $H \in \{0,1\}^{n\times n}$ *exists such that* $H^T \times_B A(F) \times_B H$ *is a strictly lower triangular (equivalently, upper triangular) matrix, where '*$\times_B$*' denotes the Boolean product.*

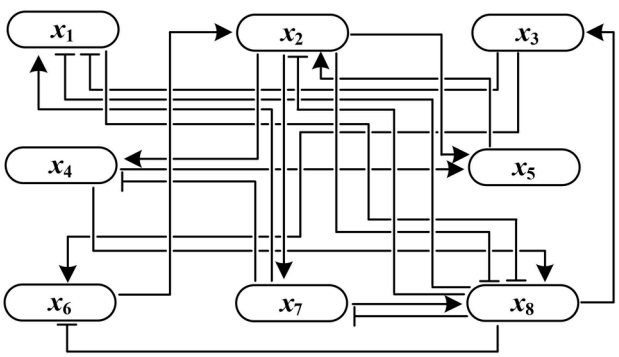


FIGURE 1 | Influence graph of $A(F)$.

$H^T \ltimes_B A(F) \ltimes_B H$ signifies that state variables of F are reordered according to H. If $H^T \ltimes_B A(F) \ltimes_B H$ is strictly lower triangular, the influence graph of F encoded by the adjacency matrix $A(F)$ turns out to be acyclic as addressed in Robert (1986). Theorem 2 and Theorem 3 stipulate that with this condition, F will converge to a unique fixed point after some iterations $k \in \mathbb{N}$. Further, $(A(F))^k = \mathbf{0}$ is a sufficient condition for global stabilization since $A(F^k) \leq (A(F))^k$ for all k.

Example 2. *Consider a BN* $F = (f_1, f_2, f_3, f_4)^T$ *with*

$$\begin{aligned} f_1(x) &= x_2 \vee x_3 \\ f_2(x) &= \neg x_4 \\ f_3(x) &= x_2 \wedge x_3 \\ f_4(x) &= \neg x_3 \end{aligned}$$

Assume that x_3 *is fixed to 1 as a control input. With* $x_3 = 1$*, The second and third iterate* F^2 *and* F^3 *are derived as*

$$F^2 = \begin{cases} x_1 := \neg x_4 \vee 1 \\ x_2 := x_3 \\ x_3 := 1 \\ x_4 := 0 \end{cases} \qquad F^3 = \begin{cases} x_1 := 1 \\ x_2 := 1 \\ x_3 := 1 \\ x_4 := 0 \end{cases}$$

Since $A(F^3) = \mathbf{0}_{4\times 4}$*, by Theorem 1 the BN globally stabilizes to* $(1,1,1,0)^T$ *in three steps from any initial state. The adjacency matrix of* F *with* $x_3 = 1$ *is*

$$A(F) = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

It is easy to compute that $(A(F))^{(4)} = \mathbf{0}_{4\times 4}$*. Hence global stability of the BN is confirmed again by Theorem 2. The latter can be proved by employing the following permutation matrix*

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

that switches the order of x_3 *and* x_4*. Since*

$$H^T \ltimes_B A(F) \ltimes_B H = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

is strictly upper triangular, by Theorem 3 $k \in \mathbb{N}$ *exists such that* $(A(F))^{(k)} = \mathbf{0}_{4\times 4}$ *(k turns out to be 4 in this case).*

For x and $P = \{(i_1, u_1), \ldots, (i_{|P|}, u_{|P|})\} \subset [n] \times \{0, 1\}$, let $\hat{x}^P$ be the state vector in which each x_{i_k} is fixed to a constant u_k, $k = 1, \ldots, |P|$. If $P = \emptyset$, $\hat{x}^\emptyset = x$. For later usage, let $P_{[n]} = \{i_1, \ldots, i_{|P|}\} \subset [n]$. $\hat{x}^P$ stands for the state vector wherein some state variables are selected as constant control inputs or are canalized by other control inputs. This notation will be utilized in developing the algorithm for global stabilization.

Assume that in a BN F, x_{i_k} has been fixed to u_k, $k = 1, \ldots, |P|$, as characterized by P. Then for x_j and f_i where $i, j \in [n] - P_{[n]}$ and $i \neq j$, x_j is called a *canalizing variable* of the transition function f_i if there exist $u, v \in \{0, 1\}$ such that setting $x_j = u$ in $\hat{x}^P$ canalizes $f_i(\hat{x}^P)$ to v. Note that all successive canalizations by $x_j = u$ are considered in checking the canalization of f_i, namely, more than one transition function may be canalized in a sequential way as the result of setting $x_j = u$. f_i is said to be a (u, v)-canalized transition function of x_j with respect to $\hat{x}^P$ [a similar definition is presented in Cheng et al. (2011a)].

To quantify the canalization effect of a state variable, denote by $C_j(P; u) \subset [n] - P_{[n]}$ the index set of all $(u, *)$-canalized transition functions of x_j with respect to $\hat{x}^P$ that are derived by setting $x_j = u$. For instance, if $x_j = u$ canalizes f_j as well as another transition function $f_{j'}$, we have $C_j(P; u) = \{j, j'\}$. It is convenient to elucidate which setting among $x_j = 0$ and $x_j = 1$ yields greater canalization effect. To this end, define

$$T_j(P) = \max(|C_j(P; 0)|, |C_j(P; 1)|)$$

as the *canalization number* of x_j with respect to $\hat{x}^P$. $T_j(P)$ equals the maximum number of canalized transition functions of x_j that are found for all state variables in $[n] - P_{[n]}$. But to describe our algorithm of global stabilization, we often need to restrict the state variables of interest to a subset of $[n] - P_{[n]}$. Formally, for $Q \subseteq [n] - P_{[n]}$ define $T_j(P, Q)$, where $j \in Q$, as

$$T_j(P, Q) = \max(|C_j(P; 0) \cap Q|, |C_j(P; 1) \cap Q|)$$

$T_j(P, Q)$ represents the maximum number of canalized transition functions of x_j that are searched only among state variables of Q.

Example 3. *Global stabilization by the control input* $x_3 = 1$ *in Example 2 can be interpreted as canalization. In view of Example 2, we can set* $P = \emptyset$ *and* $Q = \{1, 2, 3, 4\}$*. Once* x_3 *is fixed to 1,* x_4 *is also fixed to 0 in the second iterate* F^2*. Further,* x_3 *and* x_4 *canalize* x_1 *and* x_2 *to 1 in the third iterate* F^3*. Hence* $C_3(P; 1) = 3$*. Similarly,* $C_3(P; 0) = 3$ *and thus* $T_3(P, Q) = \max(3, 3) = 3$*.*

3. METHODS

3.1. Global Stabilization

Although the criterion of Theorem 3 is sufficient but not necessary, it can serve as a practical tool to determine control inputs to complex biological networks since $A(F)$ has a polynomial complexity with respect to n. Specifically, to derive $A(F)$ from the influence graph of F with n nodes, one must check whether any pair of nodes are adjacent with each other. Hence $A(F)$ is computed in $O(n^2)$. Based on Theorem 3, we now propose a scheme of deriving a set of control inputs that guarantee global stabilization of a BN F. Theorem 3 implies that if the adjacency matrix or one of its permuted matrices is strictly lower triangular, F converges to a unique fixed point. To utilize this result, we first reorder state variables of F so that the permuted adjacency matrix can be as similar to a strictly lower triangular matrix as possible. If the permuted adjacency matrix

turns out to be strictly lower triangular, no assignment of control inputs is necessary. Otherwise, we select in a sequential way a set of state variables that will be used as control inputs. In terms of the graph representation, the latter scheme is equal to making the influence graph of F acyclic by fixing some nodes, thus removing their input edges and potentially breaking cycles.

Once a state variable x_i is selected as a control input, all the incoming edges of x_i are disconnected, leading to $a_i^r = \mathbf{0}_{1\times n}$. In terms of global stabilization, we can also regard that the outgoing edges of x_i are 'disconnected' since the influence of x_i on other state variables becomes constant as does the value of x_i. Hence we will set $a_i^c = \mathbf{0}_{n\times 1}$ in our algorithm for determining control inputs. In this regard, it would be best if we first select the state variable that has the greatest outgoing edges in its upper right entries of the (permuted) adjacency matrix. Moreover, we must consider the canalization effect of the selected state variable. If the transition function of another state variable is canalized by the selected state variable, all the corresponding entries of the adjacency matrix also degenerate into zeros. How many state variables are canalized, as is quantified by the canalization number, will be also utilized as a criterion to select the control input. The following algorithm is the main result of this paper.

Algorithm 1. *Derivation of control inputs that make the adjacency matrix strictly lower triangular:*

Given a BN F with the adjacency matrix $A(F) = (a_{i,j})$, we determine a set of control inputs that ensures global stability of F. Set $P = \emptyset$ and $Q = [n]$.

1. *Permute $(a_{i,j})$ and update Q as follows.*
   a. *Sort the row vectors into an ascending order of the row vector norm. Letting $i(1), \ldots, i(n)$ be the sorted indices, we have*
   $$|a^r_{i(1)}| \leq |a^r_{i(2)}| \leq \cdots \leq |a^r_{i(n)}|$$
   b. *Permute $(a_{i,j})$ according to $i(1), \ldots, i(n)$, i.e., reorder the state variables so that $x_{i(k)}$ is placed on the kth position for all $k \in [n]$. Let $(\tilde{a}_{i,j})$ be the permuted matrix of $(a_{i,j})$.*
   c. *Set $Q = Q - \{j \in Q | d(\tilde{a}^c_j) = 0\}$.*
2. *Search for $j^* \in Q$ as follows.*
   a. *Let $K \subset Q$ be the set of indices such that*
   $$k = \arg\max_{j\in Q} d(\tilde{a}^c_j) \quad \forall k \in K$$
   b. *Among the entries of K, find j^* such that*
   $$j^* = \arg\max_{k\in K} T_k(P, Q)$$
3. *Modify $(\tilde{a}_{i,j})$ and update P and Q as follows.*
   a. *Let $u^* \in \{0, 1\}$ be the value of x_{j^*} such that $T_{j^*}(P, Q) = |C_{j^*}(P; u^*) \cap Q|$.*
   b. *Set*
   $$\tilde{a}^r_{j^*} = \tilde{a}^r_h = \mathbf{0}_{1\times n}$$
   $$\tilde{a}^c_{j^*} = \tilde{a}^c_h = \mathbf{0}_{n\times 1} \quad \forall h \in C_{j^*}(P; u^*) \cap Q$$
   c. *Update P and Q by*
   $$P = P \cup \{(j^*, u^*)\}$$
   $$Q = Q - \{j^*\} \cup C_{j^*}(P; u^*)$$
4. *If $(\tilde{a}_{i,j})$ is strictly lower triangular, terminate the algorithm. The solution to global stabilization of F is*
   $$x_{j_1} = u_1, \ldots, x_{j_{|P|}} = u_{|P|}$$
   where $P = \{(j_1, u_1), \ldots, (j_{|P|}, u_{|P|})\}$. Otherwise, return to Step 2.

In the above algorithm, P denotes the set of selected control inputs so far and Q represents eligible candidates that can be selected as control inputs in the next step ($P_{[n]} \cap Q = \emptyset$). Step 1 describes the permutation of $(a_{i,j})$ by reordering state variables. Since the permuted adjacency matrix must be akin to a strictly lower triangular one, we reorder x_i's in an ascending order of the norm of the corresponding row vectors, that is, those with more incoming edges are placed on later positions. If $d(\tilde{a}^c_j) = 0$, x_j needs not be selected as a control input since the present form of its column vector $\tilde{a}^c_j$ is already a component of a strictly lower triangular matrix. Thus x_j is removed from the candidate set Q (Step 1.c).

In Step 2, we derive the index j^* of the state variable that, if selected as a control input, can modify $(\tilde{a}_{i,j})$ so that the changed matrix approaches a strictly lower triangular matrix the most. The best candidate would be the one having the most outgoing edges in its upper right entries, which is represented by $\max d(\tilde{a}^c_j)$ (Step 2.a). If more than one state variable have the maximum $d(\tilde{a}^c_j)$, we choose the variable that has the greatest canalization number (Step 2.b), since the corresponding row and column vectors of all canalized state variables degenerate into zeros (Step 3.b), hence contributing to the modification of $(\tilde{a}_{i,j})$ to a strictly lower triangular matrix.

Once selected as a control input or canalized by any selected control input, the state variable must be excluded from the candidate set Q (Step 3.c). If the modified adjacency matrix is strictly lower triangular, the assignment of constant controls in P is the solution to global stabilization (Step 4). Otherwise, the foregoing steps are iterated until the solution is derived.

Example 4. *Using Algorithm 1, let us derive the solution to global stabilization of F in Example 1. According to Step 1.a–b of Algorithm 1, we first permute $(a_{i,j})$ by reordering state variables in an ascending order of the norm of their row vectors. The permuted adjacency matrix $(\tilde{a}_{i,j})$ is shown in* ***Table 1****. Here, $|a^r_8| = 4$, $|a^r_2| = 3$, and so on. Since $d(\tilde{a}^c_j) = 0$ for all $j \in \{3, 7, 6, 4, 5, 1\}$, we set $Q = \{2, 8\}$ by Step 1.c. This means that only the outgoing edges of x_2 and x_8 are in the upper right positions of the permuted $(\tilde{a}_{i,j})$.* ***Figure 2A*** *illustrates this topology where the outgoing edges of x_2 and x_8 are drawn in orange and green, respectively. Since $d(\tilde{a}^c_8) = 5 > d(\tilde{a}^c_2) = 3$, $j^* = 8$ is selected as the first control input by Step 2. As the value of x_8 is made constant, the incoming/outgoing edges of x_8 are removed from the influence graph as shown in* ***Figure 2B****. Referring to (1), further, f_2 is $(1, 0)$-canalized by x_8. Hence $x_8 = 1$ is*

TABLE 1 | Permuted adjacency matrix $(\tilde{a}_{i,j})$.

$\lvert a_i^r \rvert$	x_3	x_7	x_6	x_4	x_5	x_1	x_2	x_8
1	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	1
2	1	0	0	0	0	0	0	1
2	0	1	0	0	0	0	1	0
2	0	0	0	1	0	0	1	0
3	1	1	0	0	0	0	0	1
3	0	0	1	0	1	0	0	1
4	0	1	0	1	0	1	1	0

the value that maximally canalizes the remaining variables of Q—single element x_2 in this case. Applying Step 3.b, we modify $(\tilde{a}_{i,j})$ to

$$(\tilde{a}_{i,j}) = \begin{pmatrix} 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 1&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \end{pmatrix}$$

Since the above matrix is strictly lower triangular, we terminate the algorithm by determining the solution $x_8 = 1$. The resultant influence graph becomes acyclic as shown in ***Figure 2C****. The global fixed point obtained by the control input $x_8 = 1$ is $x^* = (0, 0, 1, 0, 0, 0, 0, 1)^T$, which differs from σ_1–σ_3 derived in Example 1.*

To discuss computational complexity of Algorithm 1, let $s \in \mathbb{N}$ be the maximum number of incoming edges of a node in F. In Step 1.a, sorting the row vectors needs n^2 operations in the worst case. Since Step 1.b and Step 1.c can be done in one operation, respectively, Step 1 has the maximum $n^2 + 2$ operations. Step 2.a needs n operations in the worst case. In Step 2.b, on the other hand, we need to derive the canalization number of each state variable. For a state variable with l incoming edges, we must check whether the corresponding state transition function is fixed to a constant for all 2^{l-1} combinations of arguments (one argument is the canalizing variable). Hence Step 2.b needs $n2^{s-1}$ operations in the worst case. Step 3 has four operations (Step 3.a needs two operations to determine u^*), and finally Step 4 has just one operation. Combining these factors, we conclude that Algorithm 1 can be computed in $O(n^2 + n2^{s-1})$. In other words, Algorithm 1 has polynomial complexity with respect to the number of state variables, while having exponential complexity with respect to the number of incoming edges. When the considered BN has a state variable with a huge number of incoming edges, applying Algorithm 1 may be computationally demanding. Still, Algorithm 1 is useful since it is known that BNs representing biological systems are very sparse in general—the average degree of a node is about two (Leclerc, 2008).

As duality of making a strictly lower triangular matrix, we can adjust Algorithm 1 so as to search for a set of control inputs that make the resulting adjacency matrix strictly upper triangular. To this end, we reorder state variables according to an ascending order of the column vector norm $|a_j^c|$, find the candidate control input that has the greatest $|d(a_i^r)|$, and so on. The following algorithm is analyzed in a similar way to Algorithm 1.

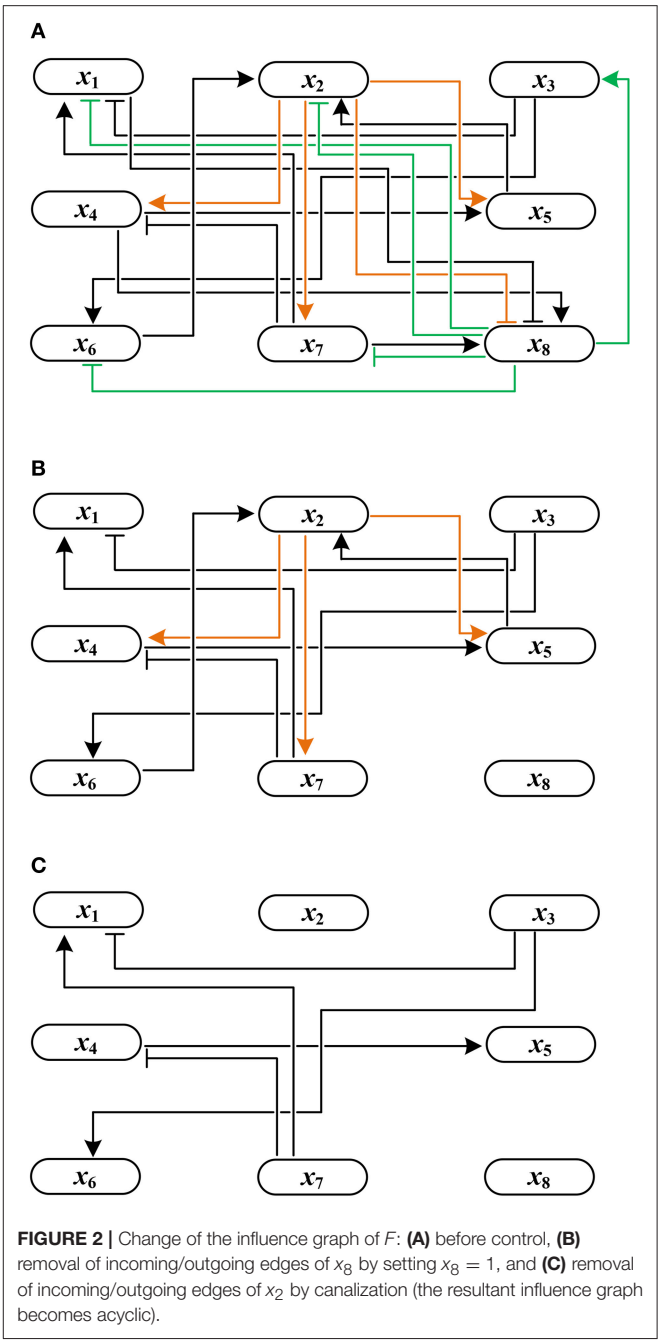


FIGURE 2 | Change of the influence graph of F: **(A)** before control, **(B)** removal of incoming/outgoing edges of x_8 by setting $x_8 = 1$, and **(C)** removal of incoming/outgoing edges of x_2 by canalization (the resultant influence graph becomes acyclic).

Algorithm 2. *Derivation of control inputs that make the adjacency matrix strictly upper triangular:*

Given a BN F with the adjacency matrix $A(F) = (a_{i,j})$, we determine a set of control inputs that ensures global stability of F. Set $P = \emptyset$ and $Q = [n]$.

1. *Permute* $(a_{i,j})$ *and update Q as follows.*
   a. *Sort the column vectors into an ascending order of the column vector norm. Letting* $j(1), \ldots, j(n)$ *be the sorted indices, we have*
   $$|a^c_{j(1)}| \leq |a^c_{j(2)}| \leq \cdots \leq |a^c_{j(n)}|$$
   b. *Permute* $(a_{i,j})$ *according to* $j(1), \ldots, j(n)$*, i.e., reorder the state variables so that* $x_{j(k)}$ *is placed on the kth position for all* $k \in [n]$*. Let* $(\tilde{a}_{i,j})$ *be the permuted matrix of* $(a_{i,j})$*.*
   c. *Set* $Q = Q - \{i \in Q | d(\tilde{a}^r_i) = 0\}$*.*
2. *Search for* $i^* \in Q$ *as follows.*
   a. *Let* $K \subset Q$ *be the set of indices such that*
   $$k = \arg\max_{i \in Q} d(\tilde{a}^r_i) \quad \forall k \in K$$
   b. *Among the entries of K, find* i^* *such that*
   $$i^* = \arg\max_{k \in K} T_k(P, Q)$$
3. *Modify* $(\tilde{a}_{i,j})$ *and update P and Q as follows.*
   a. *Let* $u^* \in \{0, 1\}$ *be the value of* x_{i^*} *such that* $T_{i^*}(P, Q) = |C_{i^*}(P; u^*) \cap Q|$*.*
   b. *Set*
   $$\tilde{a}^r_{i^*} = \tilde{a}^r_h = \mathbf{0}_{1 \times n}$$
   $$\tilde{a}^c_{i^*} = \tilde{a}^c_h = \mathbf{0}_{n \times 1} \quad \forall h \in C_{i^*}(P; u^*) \cap Q$$
   c. *Update P and Q by*
   $$P = P \cup \{(i^*, u^*)\}$$
   $$Q = Q - \{i^*\} \cup C_{i^*}(P; u^*)$$
4. *If* $(\tilde{a}_{i,j})$ *is strictly upper triangular, terminate the algorithm. The solution to global stabilization of F is*
   $$x_{i_1} = u_1, \ldots, x_{i_{|P|}} = u_{|P|}$$
   where $P = \{(i_1, u_1), \ldots, (i_{|P|}, u_{|P|})\}$*. Otherwise, return to Step 2.*

Algorithm 2 is identical to Algorithm 1 except that (i) the column vector norm $|a^c_j|$ is employed instead of the row vector norm $|a^r_i|$ in permuting the adjacency matrix (Step 1), and (ii) $d(\tilde{a}^r_i)$, the number of 1's in off-diagonal entries of a row, replaces its column counterpart $d(\tilde{a}^c_j)$ in determining the control input (Step 2). Algorithm 1 is suitable for applying to BNs in which state variables with a large number of outgoing edges produce large canalization numbers. On the other hand, Algorithm 2 is pertinent to apply to BNs where state variables with a large number of incoming edges have a tendency to have large canalization numbers.

Example 5. *Let us apply Algorithm 2 to global stabilization of F in Example 1. We first permute* $(a_{i,j})$ *according to an ascending order of the column vector norm. The permuted adjacency matrix* $(\tilde{a}_{i,j})$ *is shown in* ***Table 2****. After applying Step 2–4, we obtain* $P = \{(8, 1)\}$*, i.e.,* $x_8 = 1$ *as the control input that achieves global stabilization. Thus the control inputs and the global fixed point are the same as those derived using Algorithm 1 in Example 4.*

The proposed algorithm can be applied without modification to the case that some state variables serve as external inputs or outputs. We first remove input and output variables from the entries of the adjacency matrix. Then we derive the adjacency matrix by setting the values of external inputs and continue to apply Algorithm 1. The proposed algorithm is also applicable to the case that some state variables are disabled by mutation. For instance, if x_i is knocked out by mutation, its value is fixed to $x_i = 0$. In a similar way to Step 3.b of Algorithm 1, to deal with mutated variables we refine the adjacency matrix a priori by setting $a^r_i = a^r_h = \mathbf{0}_{1 \times n}$ and $a^c_i = a^c_h = \mathbf{0}_{n \times 1}$ for all $h \in [n]$ such that x_h is canalized by $x_i = 0$. Moreover, our algorithm can deal with the existence of uncontrollable state variables, namely those state variables that cannot be used as control inputs. Let $Q_f \subset [n]$ be the index set of uncontrollable state variables. The latter constraint can be easily implemented in the algorithm by setting $Q := [n] - Q_f$ instead of $Q = [n]$ in the initial phase.

As mentioned in Introduction, a significant advantage of the proposed algorithm is that it always guarantees a solution to global stabilization for any values of the external inputs and fixed values of mutated variables. Unless the external inputs and mutations influence the variables that are otherwise to be selected as control inputs, the algorithm gives the same solution without regard to the external inputs and mutations.

3.2. Sequential Control

Once the heterogeneity of cellular responses is eliminated by the proposed scheme of global stabilization, it would be a reasonable follow-up measure to investigate whether there is a subsequent scheme that can drive the BN further from the unique fixed point to another stable state with a desirable feature. We may realize this objective by applying various control strategies for BNs (Cheng et al., 2011a; Kim et al., 2013; Mochizuki et al., 2013). In doing so, the contribution of our study to reduce the heterogeneity of the BN strewn with many mutations and input variations will play a role as an important precedence.

In this paper, we present one of straightforward subsequent schemes—to perturb the values of external inputs after the BN reaches the unique fixed point. We first assume that among n state variables of the considered BN, m ones ($1 \leq m < n$)

TABLE 2 | Permuted adjacency matrix $(\tilde{a}_{i,j})$.

x_1	0	0	0	0	1	1	0	1
x_6	0	0	0	0	1	0	0	1
x_5	0	0	0	1	0	0	1	0
x_4	0	0	0	0	0	1	1	0
x_3	0	0	0	0	0	0	0	1
x_7	0	0	0	0	0	0	1	1
x_2	0	1	1	0	0	0	0	1
x_8	1	0	0	1	0	1	1	0
$\lvert a^c_j \rvert$	1	1	1	2	2	3	4	5

serve as external inputs, that is, they have no incoming edges in the corresponding influence graph. We also assume that some bio-markers of the cell are available to determine that the BN reaches an attractor and that a desirable phenotype turns on a specific combination of bio-markers. The proposed sequential control scheme combining global stabilization and perturbation of external inputs is addressed as follows.

Step 1: Given a BN F, apply Algorithm 1 (or Algorithm 2) to derive the set of constant control inputs $P = \{(j_1, u_1), \ldots, (j_{|P|}, u_{|P|})\}$ that ensures global stabilization of F. Find the unique fixed point $x^* \in \{0,1\}^n$ that will be reached in response to P.

Step 2: Allocating x^* as the initial state, apply all 2^m input combinations to F separately during which $|P|$ state variables $x_{j_1}, \ldots, x_{j_{|P|}}$ that were used as control inputs are set to be free variables again.

Step 3: Check whether there exists an input combination that drives F from x^* to another fixed point with the desirable phenotype.

Step 4: If no such input combination is found, the sequential control scheme fails to achieve global stabilization to a desired fixed point. Else if there are a number of input combinations that succeed in favorable global stabilization, select the minimally perturbed input combination, namely, the input combination in which the number of activated external inputs is minimum.

As elucidated in Step 4, this control strategy does not always give a solution, as the reachability of the BN starting from the unique fixed point x^* may not be expanded enough by perturbing external inputs. Nevertheless, this method is worth attempting since it is very easy to apply and computationally tractable (usually the number of external inputs m is small). Once we derive the input combination that will be applied in the second step, we can conduct the overall procedure of the sequential control scheme as follows.

1. Provide P to the considered BN.
2. Determine the convergence of the BN to x^* by observing the corresponding bio-markers. When the convergence is ensured, stop the transmission of P.
3. Engage in the second control step by providing the input combination that is derived in the preceding algorithm.
4. Confirm the convergence of the BN to the desired fixed point by observing that the bio-markers change to the corresponding values.

The practicality of the sequential control scheme will be validated in our numerical experiments.

4. APPLICATION TO BIOLOGICAL SYSTEMS

4.1. Metastasis Influence Network

To validate the practicality of the proposed algorithm, we apply it to two real biological systems. First, let us consider an influence network describing the metastatic process of cells (Cohen et al., 2015). The network graph of the metastasis influence network is shown in **Figure 3**. There are two external inputs, *ECMicroenv* and *DNADamage*, and one output, *Metastasis*. *ECMicroenv* = 1 and 0 means that the effect of the extracellular microenvironment turns on and off, respectively. *DNADamage* = 1 implies that a DNA damage occurs to the considered cell. Excluding the inputs and output, the BN of **Figure 3** has 29 free variables ($n = 29$). According to Cohen et al. (2015), it has nine possible fixed points in total.

Referring to Algorithm 1, we first compute the adjacency matrix $(a_{i,j})$ and permute it to $(\tilde{a}_{i,j})$ according to the norm of the row vector. Display of $(a_{i,j})$ and $(\tilde{a}_{i,j})$ is omitted here for space limit and the names of proteins shown in **Figure 3** will take place of the corresponding indices. By Step 2, we derive $K = \{k \in Q | k = \arg\max d(\tilde{a}_j^c)\}$ where $Q = [n] = \{1, \ldots, 29\}$. It turns out that K is a monotone set $K = \{p53\}$. Hence we have $j^* = p53$.

By Step 3.a, we replace *p53* with 0 and 1 respectively to the Boolean logic rules (**Supplementary Table S1**) to compute the canalization number. It is found that by setting *p53*=1, eight state variables *AKT1*, *AKT2*, *CTNNB1*, *NICD*, *p63*, *p73*, *SNAI1*, and *SNAI2* are maximally canalized to logic 0. For instance, the state transition equation of *AKT1* is (see **Supplementary Table S1** and Cohen et al., 2015)

$$AKT1 = CTNNB1 \wedge (NICD \vee TGFbeta \vee GF \vee CDH2) \wedge \neg p53 \wedge \neg miR34 \wedge \neg CDH1$$

Since *p53* is included in the equation in the form "$\wedge\neg p53$," $p53 = 1$ clearly leads to $AKT1 = 0$.

Further, we investigate whether other variables are subsequently canalized by these first-canalized variables and so forth. Interestingly, all the other variables are canalized to fixed values. Therefore, we find that $p53 = 1$ is a solution to global stabilization of the metastasis influence network (refer to **Supplementary Dataset S4** for a Python script of Algorithm 1 for the Metastasis influence network). To confirm our result, we use a Python package called BooleanNet (Albert et al., 2008; BooleanNet, 2018) to search for attractors of the BN with the control input $p53 = 1$ (see also the Supplementary Material **BooleanNet**). **Table 3** is the outcome of the search given every possible combination of two inputs *DNADamage* and *ECMicroenv*. We ensure the validity of the proposed scheme since **Table 3** equals the result of our scheme.

An examination of **Table 3** shows that four attractors attr1–attr4 are almost identical with each other (only the value of *TGFbeta* differs). Hence it can be said that the proposed scheme guarantees homogenous stable states of the considered BN against heterogeneity in terms of external inputs. Further, all the obtained attractors are desirable since they ensure programmed cell death and no metastasis is manifested (*Apoptosis* = 1 and *Metastasis* = 0 in all attractors). This result complies with the analysis in Cohen et al. (2015) that unless the mutation activating *NICD* and inhibiting *p53* occurs, the network will converge to apoptotic stable states.

Though any subsequent control is unnecessary in this case, we note that the proposed algorithm of global stabilization is not able to specify the features of the obtained fixed points in general since it does not use any parameter associated with the

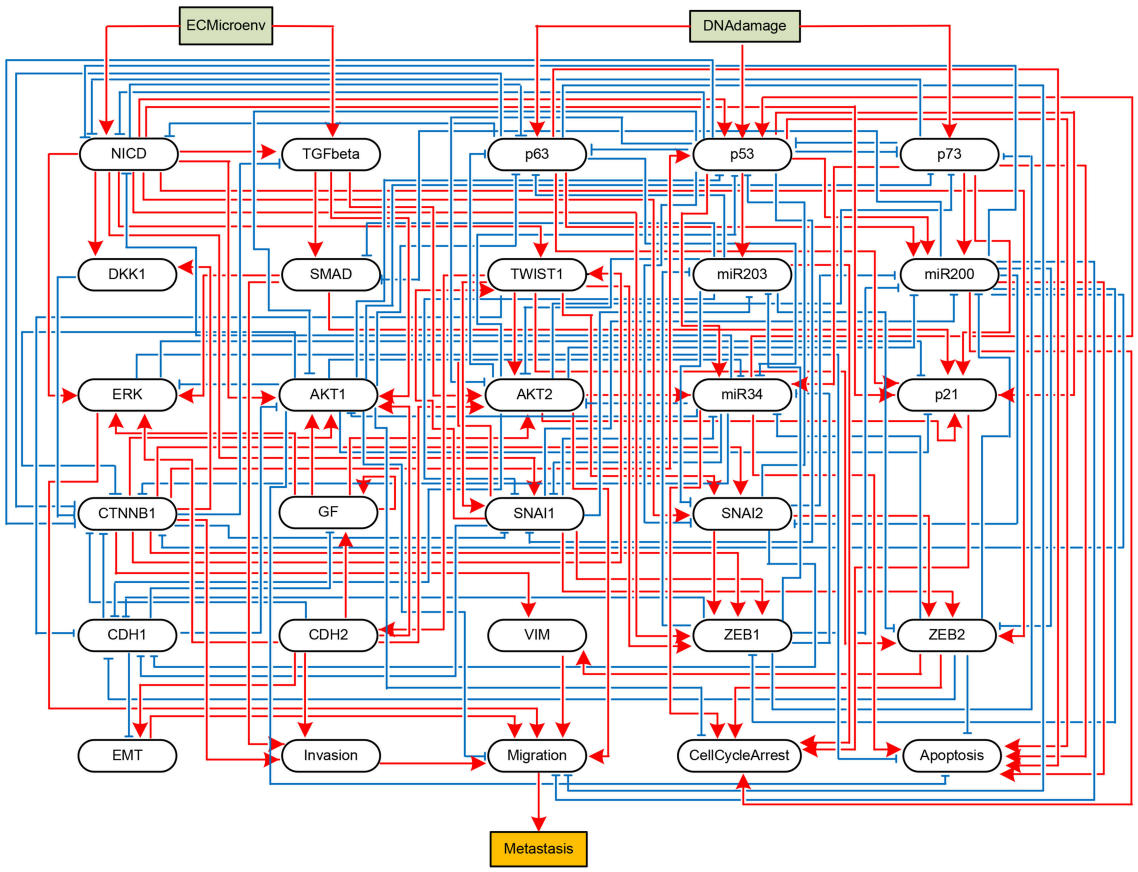


FIGURE 3 | Boolean network implementing a metastasis influence network (Cohen et al., 2015). Some nodes represent biochemical species (proteins, miRNAs, processes, etc.) and others represent phenotypes, and edges represent activating (blue) or inhibitory (red) influences of one node onto other node. The BN has two input nodes *ECMicroenv* and *DNADamage* and one output node *Metastasis*, drawn in rectangles.

desirable phenotypes. Hence, if the obtained fixed points do not have desirable features, we must apply the second control step. The latter problem will be discussed in the next case study. We also note that Algorithm 2 produces the same result $p53 = 1$ for this case study.

In Cohen et al. (2015), main consideration was devoted to constructing a logical model describing metastasis and to understanding the role of involved gene alterations. While some predictions were made on pathways and molecules triggering metastasis, no methodology was presented to determine control targets that can globally stabilize the metastasis influence network. Hence, our study can expedite further analysis of the metastasis influence network for control purposes.

4.2. MAPK Signaling Network

Next, we apply the proposed algorithm to global stabilization of the Mitogen-activated protein kinase (MAPK) signaling network that describes the mechanism underlying the influence of the MAPK signaling network on cancer cell fate decision (Grieco et al., 2013). Represented as a BN shown in **Figure 4**, the MAPK signaling network has 53 components in total, among which there are four inputs (*DNA_damage*, *EGFR_stimulus*, *FGFR3_stimulus*, and *TGFBR_stimulus*) and three outputs (*Proliferation*, *Apoptosis*, and *Growth_Arrest*).

We have applied the proposed algorithm to the MAPK signaling network with various combinations of input sets and mutation settings, which are specified in Supplementary Dataset S3 of Grieco et al. (2013). For all the possible input combinations and mutation settings, our algorithm produces the same solution set, $p38 = 1$ and $GRB2 = 1$, that globally stabilizes the considered network (both Algorithms 1 and 2 derive the same result; refer to **Supplementary Dataset S5** for a Python script of Algorithm 1). **Table 4** is a list of selected results that shows attractors with respect to five combinations of external inputs and three mutation settings, denoted by r4, r9, and r10 following Grieco et al. (2013); refer to **Supplementary Table S3** for attractors obtained with respect to all input combinations.

This results imply a remarkable virtue of the proposed algorithm, i.e., despite differences in activations of the external inputs and mutation profiles, our scheme guarantees the global stabilization of the considered network. According to **Supplementary Dataset S3** of Grieco et al. (2013), the number of attractors for each mutation setting is 3 for r4, 1 for r9, and 2 for r10. By contrast, applying the derived control inputs $p38 = 1$

TABLE 3 | Unique fixed points with $p53 = 1$.

Gene	attr1	attr2	attr3	attr4
DNADamage	**0**	**0**	**1**	**1**
ECMicroenv	**0**	**1**	**0**	**1**
AKT1	0	0	0	0
AKT2	0	0	0	0
CDH1	1	1	1	1
CDH2	0	0	0	0
CTNNB1	0	0	0	0
DKK1	0	0	0	0
ERK	0	0	0	0
GF	0	0	0	0
miR200	1	1	1	1
miR203	1	1	1	1
miR34	0	0	0	0
NICD	0	0	0	0
p21	1	1	1	1
p53	1	1	1	1
p63	0	0	0	0
p73	0	0	0	0
SMAD	0	0	0	0
SNAI1	0	0	0	0
SNAI2	0	0	0	0
TGFbeta	0	1	0	1
TWIST1	0	0	0	0
VIM	0	0	0	0
ZEB1	0	0	0	0
ZEB2	0	0	0	0
CellCycleArrest	1	1	1	1
Apoptosis	**1**	**1**	**1**	**1**
EMT	0	0	0	0
Invasion	0	0	0	0
Migration	0	0	0	0
Metastasis	**0**	**0**	**0**	**0**

The rows of two external inputs and one output, and the key gene, Apoptosis, are written in bold.

and $GRB2 = 1$, we ensure that the network converges to a fixed point for any mutation profile. Moreover, as observed in **Table 4**, the global attractor for each case of the input combination and mutation setting is very similar to one another. For instance, in the attractors for all $2^4 = 16$ input combinations (among which only five are displayed in columns 2–6 of **Table 4**), only five state variables, *ATM*, *SMAD*, *TAK1*, *TAOK*, and *TGFBR*, have different values. The attractors obtained under mutation settings also have strong similarity with each other.

The reason for this similarity is obvious. Note that the proposed algorithm always searches for the target variables and corresponding values according to the number of outgoing edges and the canalization number (Algorithms 1 and 2). Since the latter values are little influenced by perturbation of external inputs, the result will be the same in most cases. Only state variables having incoming edges from external inputs or from those variables that are directly connected with external inputs will differ. In the above case, for example, *ATM* will vary according to the input *DNA_damage* since $ATM = DNA\_damage$ (see **Supplementary Table S2**).

Once heterogeneity of cellular responses is minimized by global stabilization, we can apply further control schemes to take the derived global attractor toward another attractor with desirable features. In this numerical experiment, we try to achieve this goal by perturbing four external inputs as presented in section 3.2. An apoptotic stable state of the MAPK signaling network is characterized by $Apoptosis = Growth\_Arrest = 1$ and $Proliferation = 0$ (Grieco et al., 2013). Referring to **Table 4**, six left most attractors are apoptotic stable states while those of mutations r9 and r10 are not. We apply every combination of input perturbations to the two non-apoptotic attractors in order to conduct the second control. Note that in the second step, the foregoing control inputs $p38 = GRB2 = 1$ are not employed any more and *p38* and *GRB2* are released as free variables.

Table 5 shows the results of perturbation of external inputs after global stabilization for mutations r9 and r10 (see **Supplementary Table S4** for complete description of attractors). Note that there are a number of input combinations among 16 candidates achieving the goal, namely, invoking the BN to reach apoptotic stable states in both mutations. We select the case of $DNA\_damage = TGFBR\_stimulus = 1$ as our solution since it needs the minimum number of input perturbations. The sequential control procedure for the MAPK signaling network for mutations r4, r9, and r10 is summarized as follows (see also section 3.2).

1. Apply control inputs $p38 = GRB2 = 1$ to drive the network toward the global fixed point of each mutation setting (**Table 4** and **Supplementary Table S3**).
2. Determine the convergence of the network by observing the change of bio-markers (see Grieco et al., 2013).
3. Conduct the second control step by applying $DNA\_damage = TGFBR\_stimulus = 1$ so as to drive the network toward apoptotic attractors (**Table 5** and **Supplementary Table S4**).

Like the foregoing case study, the present result can contribute to determining control targets for the MAPK signaling network since the original study (Grieco et al., 2013) did not consider the latter topic. The major concern of Grieco et al. (2013) was to present a logical model of the MAPK signaling network and to elucidate how MAPK signaling affects cell proliferation, growth arrest, apoptotic cell death, etc. Grieco et al. (2013) applied known biological input/output data to the MAPK signaling network, based on which the underlying mechanisms were analyzed in detail. Note that such analysis was focused on understanding the mechanism with respect to feedbacks and cross-talks inherent in the model, not on determining control targets for global stabilization as done in this study.

As mentioned, the proposed algorithm cannot specify the feature of the unique fixed point in a desirable way, which may impose a burden on the second control step. But our sequential control scheme can still be useful, especially in controlling cancer cells, for the following reasons:

(i) First, for cancer cells, removing non-genetic heterogeneity via global stabilization is a very significant phase itself

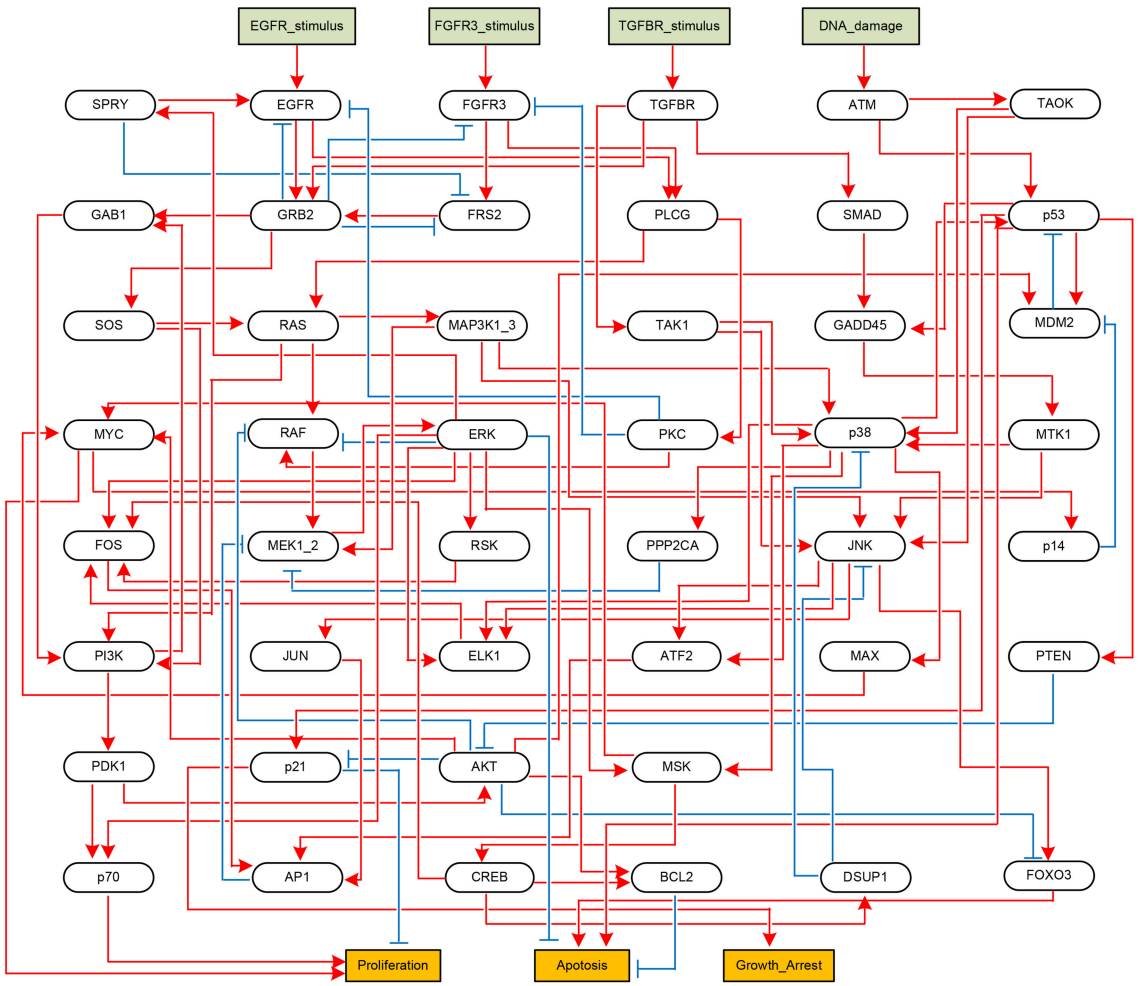


FIGURE 4 | Boolean network implementing the MAPK signaling network (Grieco et al., 2013). Each node denotes a model component. Model inputs and outputs are drawn in rectangles, and blue arrows and red T-arrows denote positive and negative regulations, respectively.

that should be achieved even though the resulting attractor is unsatisfactory. Sequential control of cancer cells, i.e., initially blocking primary mutation effects and cross-talks, and subsequently applying combinatorial targeted drugs for additional control, has been an active area of research in recent years [see, e.g., Lee et al. (2012); Vijayaraghavalu et al. (2012)].

(ii) Next, while many existing targeted drugs aim at inhibiting or activating intracellular molecules (mainly signaling proteins) of cancer cells, studies on tackling cancer cells by manipulating tumor micro-environments are also receiving a great attention. Since tumor micro-environments are characterized by external inputs in BNs, our sequential control scheme with external input control in the second step can be combined with the related methods [e.g., Bissell and Hines (2011); Quail and Joyce (2013)].

4.3. Comparative Study

To conduct a comparative study, we have applied three representative global stabilization schemes—feedback vertex set (FVS) control (Fiedler et al., 2013), the control kernel (CK) method (Kim et al., 2013), and the stable motif (SM) method (Zañudo and Albert, 2015) to the control problem of the MAPK signaling network discussed in the previous subsection.

(i) Feedback vertex set control: In graph theory, an FVS is a subset of nodes in the absence of which the digraph becomes acyclic, i.e., it contains no directed cycles (Fiedler et al., 2013; Liu and Barabàsi, 2016). Hence if constant control inputs are assigned to the state variables of an FVS, the resultant BN will eventually converge to a unique fixed point. To apply FVS control, we first identify a desired fixed point that is possessed by the considered BN, namely, a fixed point showing the desirable phenotype (*Apoptosis* = *Growth_Arrest* = 1 and *Proliferation* = 0). To this end, we randomly generated 100,000 initial states and made the MAPK signaling network evolve from each initial state. In this near brute-force searching, we found eight fixed points for r9 mutation and four ones for r10 mutation that have the desirable phenotype. We select a desired fixed point from each attractor set for r9 and r10 mutations, respectively, and derive the minimal FVS. Then we set the values of FVS according to the

TABLE 4 | Unique single attractors with *p38* = 1 and *GRB2* = 1 for various input combinations and mutation settings.

Gene	Input set to 1					Mutation settings		
	None	DNA_ damage	EGFR_ stimulus	FGFR3_ stimulus	TGFBR_ stimulus	FGFR3 = 1 (r4)	EGFR = 1 p14 = 0 (r9)	FGFR3 = 1 p14 = 0 (r10)
AKT	0	0	0	0	0	0	1	1
AP1	1	1	1	1	1	1	0	0
ATF2	1	1	1	1	1	1	1	1
ATM	0	1	0	0	0	0	0	0
Apoptosis	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**
BCL2	0	0	0	0	0	0	1	1
CREB	1	1	1	1	1	1	1	1
DUSP1	1	1	1	1	1	1	1	1
EGFR	0	0	0	0	0	0	1	0
ELK1	1	1	1	1	1	1	1	1
ERK	0	0	0	0	0	0	0	0
FGFR3	0	0	0	0	0	1	0	1
FOS	0	0	0	0	0	0	0	0
FOXO3	1	1	1	1	1	1	0	0
FRS2	0	0	0	0	0	0	0	0
GAB1	1	1	1	1	1	1	1	1
GADD45	1	1	1	1	1	1	0	0
GRB2	1	1	1	1	1	1	1	1
Growth_Arrest	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**
JNK	1	1	1	1	1	1	0	0
JUN	1	1	1	1	1	1	0	0
MAP3K1_3	1	1	1	1	1	1	1	1
MAX	1	1	1	1	1	1	1	1
MDM2	0	0	0	0	0	0	1	1
MEK1_2	0	0	0	0	0	0	0	0
MSK	1	1	1	1	1	1	1	1
MTK1	1	1	1	1	1	1	0	0
MYC	1	1	1	1	1	1	1	1
PDK1	1	1	1	1	1	1	1	1
PI3K	1	1	1	1	1	1	1	1
PKC	0	0	0	0	0	1	1	1
PLCG	0	0	0	0	0	1	1	1
PPP2CA	1	1	1	1	1	1	1	1
PTEN	1	1	1	1	1	1	0	0
Proliferation	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
RAF	1	1	1	1	1	1	0	0
RAS	1	1	1	1	1	1	1	1
RSK	0	0	0	0	0	0	0	0
SMAD	0	0	0	0	1	0	0	0
SOS	1	1	1	1	1	1	1	1
SPRY	0	0	0	0	0	0	0	0
TAK1	0	0	0	0	1	0	0	0
TAOK	0	1	0	0	0	0	0	0
TGFBR	0	0	0	0	1	0	0	0
p14	1	1	1	1	1	1	0	0
p21	1	1	1	1	1	1	0	0
p38	1	1	1	1	1	1	1	1
p53	1	1	1	1	1	1	0	0
p70	0	0	0	0	0	0	0	0

The rows of three genes composing the desirable phenotype are written in bold.

TABLE 5 | Results of perturbation of external inputs after global stabilization by $p38 = 1$ and $GRB2 = 1$.

Mutation setting	External inputs				Key genes in attractors		
	DNA_ damage	EGFR_ stimulus	FGFR3_ stimulus	TGFBR_ stimulus	Apoptosis	Growth_Arrest	Proliferation
r9	0	0	0	0	Cycle	Cycle	Cycle
	0	0	0	1	0	0	0
	0	0	1	0	Cycle	Cycle	Cycle
	0	0	1	1	0	0	0
	0	1	0	0	Cycle	Cycle	Cycle
	0	1	0	1	0	0	0
	0	1	1	0	Cycle	Cycle	Cycle
	0	1	1	1	0	0	0
	1	0	0	0	1	1	0
	1	**0**	**0**	**1**	**1**	**1**	**0**
	1	0	1	0	1	1	0
	1	0	1	1	1	1	0
	1	1	0	0	1	1	0
	1	1	0	1	1	1	0
	1	1	1	0	1	1	0
	1	1	1	1	1	1	0
r10	0	0	0	0	Cycle	Cycle	Cycle
	0	0	0	1	0	0	0
	0	0	1	0	Cycle	Cycle	Cycle
	0	0	1	1	0	0	0
	0	1	0	0	Cycle	Cycle	Cycle
	0	1	0	1	0	0	0
	0	1	1	0	Cycle	Cycle	Cycle
	0	1	1	1	0	0	0
	1	0	0	0	Cycle	Cycle	Cycle
	1	**0**	**0**	**1**	**1**	**1**	**0**
	1	0	1	0	Cycle	Cycle	Cycle
	1	0	1	1	1	1	0
	1	1	0	0	Cycle	Cycle	Cycle
	1	1	0	1	1	1	0
	1	1	1	0	Cycle	Cycle	Cycle
	1	1	1	1	1	1	0

The rows of the selected solution input combination are written in bold.

corresponding values in the selected fixed point. Although some cyclic attractors also have the desirable phenotype, we did not use them for the purpose of focusing on fixed points.

Referring to **Supplementary Dataset S1**, we found that Attractor 2 of r9 mutation and Attractor 21 of r10 mutation are the same. Hence by selecting this fixed point, we can achieve global stabilization of the BN with desirable phenotype irrespective of the existence of both r9 and r10 mutations. **Table 6** shows the minimal FVSs that take the MAPK signaling network toward the desired fixed point. The result of **Table 6** is similar to that of **Table 5** in that both solve the global stabilization problem by activating two external inputs (*DNA_damage* and *TGFBR_stimulus*) and by setting some state variables to be constant controls.

In term of accessibility of the modeling information, FVS control is superior since it does not need the exact Boolean logic of the BN. On the other hand, the solution of the proposed scheme is more efficient in this numerical experiment since the number of control inputs is less than that of FVS control. In fact, our solution set $p38 = 1$ and $GRB2 = 1$ is included in the minimal FVS as seen in **Table 6**.

(ii) Control kernel method: In Kim et al. (2013), the control kernel is defined as the minimal set of nodes that need to be regulated to drive the network to converge to a desired attractor for all initial states. A genetic algorithm (GA) is employed to find the minimal set among randomly selected candidate node sets. Following the method addressed in Kim et al. (2013), we found the control kernel that drives the MAPK signaling network to a fixed point with the desirable phenotype (*Apoptosis* = *Growth_Arrest* = 1 and *Proliferation* = 0). Since global stabilization must be valid for either r9 or r10 mutation, we searched for the control kernel for each mutation case separately

and extracted common control kernels, if any. The control kernel method usually chooses a desired fixed point, based on which an appropriate control kernel is explored. But we did not specify any desired fixed point in this case study. Instead, we adapted the control kernel algorithm and discovered feasible control kernels and their corresponding fixed points yielding the desirable phenotype simultaneously in the search space of the control kernel method.

It is found that no control kernel having size one exists that achieves global stabilization of the BN. With the size of the control kernel set to be two, we found nine control kernels that solve the control problem, as shown in **Table 7** and **Supplementary Dataset S2**. Interestingly, our solution set $p38 = 1$ and $GRB2=1$ is not included in the derived control kernels. This is due to the property of our adapted searching algorithm that it explores the control kernel and a desired attractor in one single step. The result of **Table 7** indicates that the control kernel has superior performance than the proposed scheme since it does not need any external input to be activated. In term of computational load, however, our algorithm is much better since while the control kernel method takes more than 72 h to obtain the result, our algorithm yields the control inputs and the associated external inputs in a few minutes.

(iii) Stable motif method: A stable motif is referred to as a set of nodes and their corresponding states such that the nodes form a minimal strongly connected component and their states form a partial fixed point of the BN (Zañudo and Albert, 2015). Stable motifs can be regarded as control targets since once they reach certain Boolean values, they are preserved against other updating schemes due to their dynamical property of being partial fixed points. In Zañudo and Albert (2015), the set of stable motifs is first computed, followed by reducing the number of control targets in the stable motif using the *stable motif control algorithm*. The stable motif method is remarkable since it is the first network control approach that combines the structural and functional information of Boolean networks to determine control inputs for stabilization.

We have applied the stable motif method to controlling the MAPK signaling network that is influenced by r9 and r10 mutations. The stable motif control algorithm was implemented based on the method of Zañudo et al. (2017), and StableMotifs java library devised in Zañudo and Albert (2015) was used to realize the simulation code. It is found that the set of stable motifs for each mutation profile contains more than 10 state variables. However, through the stable motif control algorithm, we derived a number of stable motif control sets consisting of only four external inputs (**Supplementary Dataset S3**). Among them, four combinations shown in **Table 8** globally stabilize the BN to a fixed point having the desirable phenotype ($Apoptosis = Growth\_Arrest = 1$ and $Proliferation = 0$) for both r9 and r10 mutations.

TABLE 8 | Stable motif control sets for control of the MAPK signaling network with r9 and r10 mutations (refer to **Supplementary Dataset S3** for associated desired fixed points).

	External inputs			
	DNA_damage	**EGFR_stimulus**	**FGFR3_stimulus**	**TGFBR_stimulus**
1	1	0	0	1
2	1	0	1	1
3	1	1	0	1
4	1	1	1	1

TABLE 6 | Minimal FVS for control of the MAPK signaling network with r9 and r10 mutations (refer to **Supplementary Dataset S1** for associated desired fixed points).

External inputs				Internal variables					
DNA_damage	**EGFR_stimulus**	**FGFR3_stimulus**	**TGFBR_stimulus**	**ERK**	**p53**	**p38**	**PKC**	**GRB2**	**GAB1**
1	0	0	1	0	1	1	1	1	1

TABLE 7 | Control kernels with size two for control of the MAPK signaling network with r9 and r10 mutations (refer to **Supplementary Dataset S2** for associated desired fixed points).

	External inputs				Internal variables					
	DNA_damage	**EGFR_stimulus**	**FGFR3_stimulus**	**TGFBR_stimulus**	**ATM**	**FRS2**	**GRB2**	**TGFBR**	**p53**	**MDM2**
1	-	-	-	-	1	1	-	-	-	-
2	-	-	-	-	1	-	1	-	-	-
3	-	-	-	-	1	-	-	1	-	-
4	-	-	-	-	-	1	-	-	-	1
5	-	-	-	-	-	1	-	-	1	-
6	-	-	-	-	-	-	1	-	-	1
7	-	-	-	-	-	-	1	-	1	-
8	-	-	-	-	-	-	-	1	-	1
9	-	-	-	-	-	-	-	1	1	-

"-" indicates that the corresponding variable or external input is not needed.

TABLE 9 | Comparison between the proposed scheme and feedback vertex set, control kernel, and stable motif methods that are applied to controlling the MAPK signaling network.

	Proposed scheme	Feedback vertex set	Control kernel	Stable motif
Find control targets for global stabilization	Yes	Yes	Yes	Yes
Applicable to large-scale BNs ($n \geq 100$)	Yes	Yes	Yes†	No
Need to know Boolean logic of the network	Yes	No	Yes	Yes
Procedure	1. Global stabilization by the adjacency matrix 2. Determine external inputs to steer the BN toward a desired attractor	1. Find FVSs using network topology 2. Fix values of FVS states corresponding to the desired attractor	Check whether the BN can be steered toward the desired attractor by brute-force method (sample initial states for large networks)	1. Compute stable motifs 2. Derive optimal stable motif nodes that take the BN to the desired attractor

†*Note that the control kernel method is computationally intractable, if not impossible, for large-scale BNs since it takes huge time to find control kernels for BNs with large n.*

The result of the stable motif method is efficient in that the solution set includes no internal variables. Hence it would be more advantageous when manipulating control targets in biological experiments. On the other hand, it is necessary to identify all attractors of the MAPK signaling network before determining the stable motif control set that will be utilized as actual controls since some quasi-attractors induced in the network reduction procedure are not real attractors of the BN (see **Supplementary Dataset S3**).

The primary difference between the proposed scheme and the existing methods is that the proposed scheme is a purely analytical approach for solving the global stabilization problem based on structural and algebraic information of the BN. The proposed scheme is particularly useful for large-scale biological networks as it does not involve any numerical search algorithm with demanding complexity. **Table 9** summarizes our comparative study with a brief review of the procedure of each control scheme.

5. CONCLUSION

The problem of global stabilization of BNs has been addressed in this paper to control the heterogeneous cellular behavior for homogeneous responses. We have proposed an algorithm determining a set of constant control inputs that can drive the controlled BN to an unspecified global fixed point. A subsequent control to transform the fixed point to a desired attractor is further presented using perturbation of external inputs. The proposed sequential control method is practical in that the procedure of selecting control inputs is simple and has polynomial computational complexity with respect to the dimension of state variables, while having exponential complexity with respect to in-degree of BNs. In addition, the proposed method can be used for any combination of external inputs and mutations. The results of numerical experiments on the metastasis regulation influence network and MAPK signaling network demonstrate the applicability of the proposed control scheme. Furthermore, our experimental studies show that the proposed sequential control can drive the BN to reach a desired final attractor and the proposed global stabilization can be utilized as a preparatory step.

AUTHOR CONTRIBUTIONS

J-MY devised the algorithm and implemented it. C-KL worked on simulation analysis. K-HC designed the project and supervised the study. J-MY and K-HC wrote the manuscript.

FUNDING

This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korea Government, the Ministry of Science and ICT (2017R1A2A1A17069642, 2015M3A9A7067220 and 2013M3A9A7046303) and also by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2015R1D1A1A01056764).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphys.2018.00774/full#supplementary-material

Supplementary information on the two biological Boolean network models, the details of the proposed control targets, and the implementation information and results of comparative studies are provided:

Supplementary Table S1 | Boolean logical rules describing the activity of nodes in the Metastasis influence network.

Supplementary Table S2 | Boolean logical rules describing the activity of nodes in the MAPK signaling network.

Supplementary Table S3 | Unique single attractors with $p38 = 1$ and $GRB2 = 1$ for all the possible input combinations in the MAPK signaling network.

Supplementary Table S4 | Complete description of the attractors in **Table 4** that are obtained by perturbation of external inputs after global stabilization by $p38 = 1$ and $GRB2 = 1$.

Supplementary Dataset S1 | Results of FVS control for the MAPK signaling network.

- "r9_RealAttractors" sheet: All the attractors of the BN with r9 mutation that are obtained by randomly generating 100,000 initial states.
- "r10_RealAttractors" sheet: All the attractors of the BN with r10 mutation that are obtained by randomly generating 100,000 initial states.
- "Minimal FVSs" sheet: Derived minimal FVSs with respect to Attractor 2 in r9_RealAttractors and Attractor 21 in r10_RealAttractors sheets.
- "Results" sheet: Selected minimal FVSs and the desired fixed points for r9 and r10 mutations.

Supplementary Dataset S2 | Results of the control kernel method for the MAPK signaling network.

- "r9_CKs" sheet: All the control kernels with size two that stabilize the BN with r9 mutation to a set of desired fixed points.
- "r10_CKs" sheet: All the control kernels with size two that stabilize the BN with r10 mutation to a set of desired fixed points.
- "Results" sheet: Intersection of the control kernels with size two for r9 and r10 mutations.

Supplementary Dataset S3 | Results of the stable motif method for the MAPK signaling network.

- "r9_RealAttractors" sheet: All the attractors of the BN with r9 mutation that are obtained by randomly generating 100,000 initial states.
- "r10_RealAttractors" sheet: All the attractors of the BN with r10 mutation that are obtained by randomly generating 100,000 initial states.
- "r9_StableMotifControlSets" sheet: Stable motif control sets of the BN with r9 mutation that are obtained by the stable motif control algorithm.
- "r10_StableMotifControlSets" sheet: Stable motif control sets of the BN with r10 mutation that are obtained by the stable motif control algorithm.
- "r9_and_r10_QuasiAttractors" sheet: Quasi-attractors of the BN with r9 and r10 mutations that are obtained by the stable motif control algorithm.
- "r9_StableMotifControl_results" sheet: Desired fixed points of the BN with r9 mutation that are obtained by applying common stable motif control sets.
- "r10_StableMotifControl_results" sheet: Desired fixed points of the BN with r10 mutation that are obtained by applying common stable motif control sets.

Supplementary Dataset S4 | A Python script that conducts global stabilization by Algorithm 1 for the Metastasis influence network (section 4.1). It can be also downloaded from https://github.com/choonlog/Global-stabilization/tree/master/GS_Adjacency_Matrix.

Supplementary Dataset S5 | A Python script that conducts global stabilization by Algorithm 1 for the MAPK signaling network (section 4.2). It can be also downloaded from https://github.com/choonlog/Global-stabilization/tree/master/GS_Adjacency_Matrix.

BooleanNet | A Python package called BooleanNet executed on Python 3.5 is used in searching for attractors in all the numerical experiments. By setting simulation parameters of the initial state of each node, logical functions of the BN, limit of trajectory steps, and the update scheme, we can find associated attractors. Notice that we have modified the source code of BooleanNet for enhancing computing efficiency. Download distribution is provided in https://github.com/choonlog/Global-stabilization.

REFERENCES

Akutsu, T., Kuhara, S., Maruyama, O., and Miyano, S. (1998). A system for identifying genetic networks from gene expression patterns produced by gene disruptions and overexpressions. *Genome Informatics* 9, 151–160.

Albert, I., Thakar, J., Li, S., Zhang, R., and Albert, R. (2008). Boolean network simulations for life scientists. *Source Code Biol. Med.* 3:16. doi: 10.1186/1751-0473-3-16

Biane, C., and Delaplace, F. (2017). "Abduction based drug target discovery using Boolean control network," in *Computational Methods in Systems Biology (CMSB 2017). Lecture Notes in Computer Science, 10545*, eds J. Feret and H. Koeppl (Cham: Springer), 57–73.

Bissell, M. J., and Hines, W. C. (2011). Why don't we get more cancer? A proposed role of the microenvironment in restraining cancer progression. *Nat. Med.* 17, 320–329. doi: 10.1038/nm.2328

BooleanNet. (2018). Available online at: https://github.com/ialbert/booleannet

Brock, A., Chang, H., and Huang, S. (2009). Non-genetic heterogeneity–a mutation-independent driving force for the somatic evolution of tumours. *Nat. Rev. Genet.* 10, 336–342. doi: 10.1038/nrg2556

Burrell, R. A., McGranahan, N., Bartek, J., and Swanton, C. (2013). The causes and consequences of genetic heterogeneity in cancer evolution. *Nature* 501, 338–345. doi: 10.1038/nature12625

Cheng, D., Qi, H., and Li, Z. (2011a). *Analysis and control of Boolean networks—a semi-tensor product approach.* London: Springer-Verlag.

Cheng, D., Qi, H., Li, Z., and Liu, J. B. (2011b). Stability and stabilization of Boolean networks. *Int. J. Rob. Nonlinear Control* 21, 134–156. doi: 10.1002/rnc.1581

Cheng, X., Qiu, Y., Hou, W., and Ching, W. K. (2017). Integer programming-based method for observability of singleton attractors in Boolean networks. *IET Syst. Biol.* 11, 30–35. doi: 10.1049/iet-syb.2016.0022

Cohen, D. P., Martignetti, L., Robine, S., Barillot, E., Zinovyev, A., and Calzone, L. (2015). Mathematical modelling of molecular pathways enabling tumour cell invasion and migration. *PLoS Comput. Biol.* 11:e1004571. doi: 10.1371/journal.pcbi.1004571

Cornelius, S. P. , Kath, W. L., and Motter, A. E. (2013). Realistic control of network dynamics, *Nat. Commun.* 4:1942. doi: 10.1038/ncomms2939

Dagogo-Jack, I., and Shaw, A. T. (2018). Tumour heterogeneity and resistance to cancer therapies. *Nat. Rev. Clin. Oncol.* 15, 81–94. doi: 10.1038/nrclinonc.2017.166

Fiedler, B., Mochizuki, A., Kurosawa, G., and Saito, D., (2013). Dynamics and control at feedback vertex sets. I: informative and determining nodes in regulatory networks. *J. Dyn. Differ. Equ.* 25, 563–604. doi: 10.1007/s10884-013-9312-7

Gonzalez, A. G., Naldi, A., Sánchez, L., Thieffry, D., and Chaouiya, C. (2006). GINsim: a software suite for the qualitative modelling, simulation and analysis of regulatory networks. *Biosystems* 84, 91–100. doi: 10.1016/j.biosystems.2005.10.003

Grieco, L., Calzone, L., Bernard-Pierrot, I., Radvanyi, F., Kahn-Perlès, B., and Thieffry, D. (2013). Integrative modelling of the influence of MAPK network on cancer cell fate decision. *PLoS Comput. Biol.* 9:e1003286. doi: 10.1371/journal.pcbi.1003286

Helikar, T., and Rogers, J. A. (2009). ChemChains: a platform for simulation and analysis of biochemical networks aimed to laboratory scientists. *BMC Syst. Biol.* 3:58. doi: 10.1186/1752-0509-3-58

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.* 2, 437–467. doi: 10.1016/0022-5193(69)90015-0

Kauffman, S., Peterson, C., Samuelsson, B., and Troein, C. (2004). Genetic networks with canalyzing Boolean rules are always stable. *Proc. Natl. Acad. Sci. U.S.A.* 101, 17102–17107. doi: 10.1073/pnas.0407783101

Kim, S., Kim, J., and Cho, K. H. (2007). Inferring gene regulatory networks from temporal expression profiles under time-delay and noise. *Comput. Biol. Chem.* 31, 239–245. doi: 10.1016/j.compbiolchem.2007.03.013

Kim, J., Park, S. M., and Cho, K. H. (2013). Discovery of a kernel for controlling biomolecular regulatory networks. *Sci. Rep.* 3:2223. doi: 10.1038/srep02223

Leclerc, R. D. (2008). Survival of the sparsest: robust gene networks are parsimonious. *Mol. Syst. Biol.* 4:213. doi: 10.1038/msb.2008.52

Lee, M. J., Ye, A. S., Gardino, A. K., Heijink, A. M., Sorger, P. K., MacBeath, G., et al. (2012). Sequential application of anticancer drugs enhances cell death by rewiring apoptotic signaling networks. *Cell* 149, 780–794. doi: 10.1016/j.cell.2012.03.031

Liu, Y. Y., and Barabàsi, A. L. (2016). Control principles of complex systems. *Rev. Modern Phys.* 88:035006. doi: 10.1103/RevModPhys.88.035006

Liu, Y. Y., Slotine, J. J., and Barabási, A. L. (2011). Controllability of complex networks. *Nature* 473, 167–173. doi: 10.1038/nature10011

McGranahan, N., and Swanton, C. (2017). Clonal heterogeneity and tumor evolution: past, present, and the future. *Cell* 168, 613–628. doi: 10.1016/j.cell.2017.01.018

Mochizuki, A., Fiedler, B., Kurosawa, G., and Saito, D. (2013). Dynamics and control at feedback vertex sets. II: A faithful monitor to determine the diversity of molecular activities in regulatory networks. *J. Theor. Biol.* 335, 130–146. doi: 10.1016/j.jtbi.2013.06.009

Mroz, E. A., Tward, A. M., Hammon, R. J., Ren, Y., and Rocco, J. W. (2015). Intra-tumor genetic heterogeneity and mortality in head and neck cancer: analysis of data from the Cancer Genome Atlas. *PLoS Med.* 12:e1001786. doi: 10.1371/journal.pmed.1001786

Murray, P. J., Kang, J. W., Mirams, G. R., Shin, S. Y., Byrne, H. M., Maini, P. K., et al. (2010). Modelling spatially regulated β-catenin dynamics and invasion in intestinal crypts. *Biophys. J.* 99, 716–725. doi: 10.1016/j.bpj.2010.05.016

Park, S. G., Lee, T., Kang, H. Y., Park, K., Cho, K. H., and Jung, G. (2006). The influence of the signal dynamics of activated form of IKK on NF-κB and anti-apoptotic gene expressions: a systems biology approach. *FEBS Lett.* 580, 822–830. doi: 10.1016/j.febslet.2006.01.004

Paulevé, L., and Richard, A. (2012). Static analysis of Boolean networks based on interaction graphs: a survey. *Electron. Notes Theor. Comput. Sci.* 284, 93–104. doi: 10.1016/j.entcs.2012.05.017

Quail, D. F., and Joyce, J. A. (2013). Microenvironmental regulation of tumor progression and metastasis. *Nat. Med.* 19, 1423–1437. doi: 10.1038/nm.3394

Robert, F. (1986). *Discrete Iterations: A Metric Study*. Berlin: Springer-Verlag.

Shaffer, S. M., Dunagin, M. C., Torborg, S. R., Torre, E. A., Emert, B., Krepler, C., et al. (2017). Rare cell variability and drug-induced reprogramming as a mode of cancer drug resistance. *Nature* 546, 431–435. doi: 10.1038/nature22794

Vijayaraghavalu, S., Dermawan, J. K., Cheriyath, V., and Labhasetwar, V. (2012). Highly synergistic effect of sequential treatment with epigenetic and anticancer drugs to overcome drug resistance in breast cancer cells is mediated via activation of p21 gene expression leading to G2/M cycle arrest. *Mol. Pharm.* 10, 337–352. doi: 10.1021/mp3004622

Wang, L. Z., Su, R. Q., Huang, Z. G., Wang, X., Wang, W. X., Grebogi, C., et al. (2016). A geometrical approach to control and controllability of nonlinear dynamical networks. *Nat. Commun.* 7:11323. doi: 10.1038/ncomms11323

Zañudo, J. G. T., and Albert, R. (2015). Cell fate reprogramming by control of intracellular network dynamics. *PLoS Comput. Biol.* 11:e1004193. doi: 10.1371/journal.pcbi.1004193

Zañudo, J. G. T., Yang, G., and Albert, R. (2017). Structure-based control of complex networks with nonlinear dynamics. *Proc. Natl. Acad. Sci. U.S.A.* 114, 7234–7239. doi: 10.1073/pnas.1617387114

Zheng, Q., Shen, L., Shang, X., and Liu, W. (2016). Detecting small attractors of large Boolean networks by function-reduction-based strategy. *IET Syst. Biol.* 10, 49–56. doi: 10.1049/iet-syb.2015.0027

Zheng, D., Yang, G., Li, X., Wang, Z., Liu, F., He, L., et al. (2013). An efficient algorithm for computing attractors of synchronous and asynchronous Boolean networks. *PLoS ONE* 8:e60593. doi: 10.1371/journal.pone.0060593

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

ORIGINAL RESEARCH
published: 02 August 2018
doi: 10.3389/fphys.2018.00878

A Mechanistic Computational Model Reveals That Plasticity of CD4$^+$ T Cell Differentiation Is a Function of Cytokine Composition and Dosage

Bhanwar Lal Puniya[1], Robert G. Todd[2], Akram Mohammed[1], Deborah M. Brown[3,4], Matteo Barberis[5,6]* and Tomáš Helikar[1]**

[1] Department of Biochemistry, University of Nebraska–Lincoln, Lincoln, NE, United States, [2] Department of Natural and Applied Sciences, Mount Mercy University, Cedar Rapids, IA, United States, [3] School of Biological Sciences, University of Nebraska–Lincoln, Lincoln, NE, United States, [4] Nebraska Center for Virology, University of Nebraska–Lincoln, Lincoln, NE, United States, [5] Synthetic Systems Biology and Nuclear Organization, Swammerdam Institute for Life Sciences, University of Amsterdam, Amsterdam, Netherlands, [6] Molecular Cell Physiology, VU University Amsterdam, Amsterdam, Netherlands

OPEN ACCESS

Edited by:
Doron Levy,
University of Maryland, College Park, United States

Reviewed by:
Kyle B. Gustafson,
Naval Surface Warfare Center Carderock Division (NSWCCD), United States
Alexey Goltsov,
Abertay University, United Kingdom

***Correspondence:**
Robert G. Todd
rtodd@mtmercy.edu
Matteo Barberis
matteo@barberislab.com
Tomáš Helikar
thelikar2@unl.edu

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: 09 December 2017
Accepted: 19 June 2018
Published: 02 August 2018

Citation:
Puniya BL, Todd RG, Mohammed A, Brown DM, Barberis M and Helikar T (2018) A Mechanistic Computational Model Reveals That Plasticity of CD4$^+$ T Cell Differentiation Is a Function of Cytokine Composition and Dosage. Front. Physiol. 9:878. doi: 10.3389/fphys.2018.00878

CD4$^+$ T cells provide cell-mediated immunity in response to various antigens. During an immune response, naïve CD4$^+$ T cells differentiate into specialized effector T helper (Th1, Th2, and Th17) cells and induced regulatory (iTreg) cells based on a cytokine milieu. In recent studies, complex phenotypes resembling more than one classical T cell lineage have been experimentally observed. Herein, we sought to characterize the capacity of T cell differentiation in response to the complex extracellular environment. We constructed a comprehensive mechanistic (logical) computational model of the signal transduction that regulates T cell differentiation. The model's dynamics were characterized and analyzed under 511 different environmental conditions. Under these conditions, the model predicted the classical as well as the novel complex (mixed) T cell phenotypes that can co-express transcription factors (TFs) related to multiple differentiated T cell lineages. Analyses of the model suggest that the lineage decision is regulated by both compositions and dosage of signals that constitute the extracellular environment. In this regard, we first characterized the specific patterns of extracellular environments that result in novel T cell phenotypes. Next, we predicted the inputs that can regulate the transition between the canonical and complex T cell phenotypes in a dose-dependent manner. Finally, we predicted the optimal levels of inputs that can simultaneously maximize the activity of multiple lineage-specifying TFs and that can drive a phenotype toward one of the co-expressed TFs. In conclusion, our study provides new insights into the plasticity of CD4$^+$ T cell differentiation, and also acts as a tool to design testable hypotheses for the generation of complex T cell phenotypes by various input combinations and dosages.

Keywords: CD4$^+$ T cell differentiation, T cell plasticity, complex T cell phenotypes, regulation of T cell plasticity, cytokine compositions, cytokine dosage

INTRODUCTION

The diversity and number of immunity-related diseases require a high level of heterogeneity in the immune system to maintain the overall well-being of a human. Early studies of immune responses led to a discovery that the $CD4^+$ T cells (referred to as T cells), which are critical players in immunity, can be classified into two subtypes - T helper 1 (Th1) and T helper 2 (Th2) cells (Mosmann et al., 1986). Each type of effector T cell produces a specific set of cytokines that define the function of the cell and the way it further governs the immune response. Specifically, the Th1 cells are responsible for several autoimmune diseases, whereas the Th2 cells are the mediators in cases of allergy and asthma (Reiner, 2007; Zhu and Paul, 2008). More recently, a number of additional T cell subtypes, including the inducible regulatory T cells (iTregs) (Groux et al., 1997; Chen et al., 2003; Schmitt and Williams, 2013), T helper 17 (Th17) (Romagnani, 2000; Harrington et al., 2005; Mangan et al., 2006), T helper 9 (Th9) (Dardalhon et al., 2008; Veldhoen et al., 2008; Soroosh and Doherty, 2009), and follicular T helper cells (Tfh) (Breitfeld et al., 2000; Schaerli et al., 2000) have been discovered, and their functions have been extensively studied. For example, the Th17 cells have been found to be responsible for assisting the immune response against extracellular bacteria and fungi, whereas the main role of the iTregs is to maintain the balance and regulate immune responses by the T helper cell subtypes (Zhu and Paul, 2008). The Th9 cells have been found to be involved in pathogen immunity and inflammatory diseases (Kaplan, 2013). Finally, the Tfh cells assist in T cell-dependent B cell response (Breitfeld et al., 2000; Schaerli et al., 2000; Ma et al., 2012).

In addition, recent studies suggest that some T helper cells are capable of switching and exhibiting phenotypes of one of the alternative effector T cells, depending on the combination of input signals that the cell receives. For example, the iTregs and Th17 can switch from one phenotype to the other in response to the pleiotropic cytokine interleukin-6 (IL-6) (Xu et al., 2007; Lee et al., 2009a; Rowell and Wilson, 2009; Kimura and Kishimoto, 2010). The fully differentiated Th17 cells have been observed to produce Th1-cell-specific cytokines (Shi et al., 2008; Lee et al., 2009b; Nindl et al., 2012; Harbour et al., 2015). The Th2 cells have been reported to further develop into Th9 cells (Veldhoen et al., 2008). More complexity in T cell differentiation was observed in the form of co-expression of mutually exclusive lineage-specifying transcription factors (TFs) (Peine et al., 2013; Bock et al., 2017). This co-expression can lead to the development of stable or intermediate subtypes that share characteristics of more than one type of T cell (Tartar et al., 2010). Examples of such mixed (complex) phenotypes include Th1–Th2 (Peine et al., 2013; Bock et al., 2017) and Th1–Th17 (Kullberg et al., 2006; Morrison et al., 2013).

The differentiation process is governed by the regulation of multiple, mutually cross-linked signaling pathways, which form complex networks (Zhu et al., 2010). The stimulation of the naive $CD4^+$ T cells by various cytokines triggers a cascade of signaling events, such as the activation of the JAK/STAT pathways that lead to the activation of T cell lineage-specifying TFs (Murphy and Reiner, 2002; Kaiko et al., 2008). For example, the commitment to Th1 lineage is initiated through signaling by interferon gamma (IFN-γ) and IL-12, leading to the activation of STAT1/STAT4, which in turn activate the T box expressed in T cells (Tbet). Differentiation into Th2 is stimulated by the activation of the GATA binding protein 3 (GATA3) TF through STAT6 signaling. The differentiation of naive T cells into Th17 is governed by the retinoic acid receptor-related orphan receptor gamma t (RORγt) TF, and by the cytokines i.e., IL-6, IL-21, IL-23 and the transforming growth factor beta (TGF-β) (Aggarwal et al., 2003; Harrington et al., 2005; Park et al., 2005; Tesmer et al., 2008). In addition, the TGF-β inhibits T cell differentiation to both the Th1 and Th2 lineages and is also conducive to the cell's commitment to the iTregs lineage (Schmitt and Williams, 2013).

The complexity of biochemical networks underlying the regulation of T cell differentiation leads to additional questions regarding the mechanisms of the immune response. For instance, based on a large number of possible combinations of extracellular cues, we may ask the following questions: (i) How does the cell decide into which subsequent lineage to differentiate? (ii) What specific combinations of signals are driving a possible switch to a different lineage? (iii) What specific mechanisms are responsible for the T cell differentiation capacity and plasticity?

While regulation of T cell differentiation in the context of the diverse cytokine microenvironment has been studied extensively, effects of the interplay among multiple cytokines on T cell differentiation remain an open question. A systems-level computational model can be used to explore whether, and to what extent, the extracellular cytokine milieu affects the T cell differentiation program. Recently, computational models using various types of mathematical approaches investigated the regulation of phenotypic plasticity, and dynamics in response to diseases (Naldi et al., 2010; Carbo et al., 2013, 2014; Abou-Jaoudé et al., 2014; Martinez-Sanchez et al., 2015). Predictions from these models include novel T cell differentiation pathways (Naldi et al., 2010), transition among T cell types under various micro-environments and perturbations (Martinez-Sanchez et al., 2015), peroxisome proliferator-activated receptor gamma-dependent regulation of Th17 to iTreg switch (Carbo et al., 2013), and IL-21-dependent modulation of IL-10 (Carbo et al., 2014).

Here, we explored the effect of the interplay among extracellular cytokines on differentiation of T cells and their plasticity. We have developed a logic-based computational model (Helikar and Rogers, 2009; Helikar et al., 2012a,b, 2013; Naldi et al., 2015; Abou-Jaoudé et al., 2016; Barberis et al., 2017; Linke et al., 2017) of a signal transduction network that regulates the differentiation process of naive T cells to Th1, Th2, Th17, and iTreg cells and analyzed its dynamics. Local protein–protein regulatory information was manually curated to construct the mechanistic model that contains lineage-specifying TFs (Tbet, GATA3, RORγt, and Foxp3), various signal transducers and activators of transcription (STATs), and other signaling molecules. The model consists of 96 regulatory interactions among 38 components. To explore the entire cytokine microenvironment, we analyzed the model's dynamics under (i) all possible combinations of extracellular signals, and (ii) various input dosages. The analysis of the model resulted in dynamic signatures that represent previously described, as

well as novel cellular phenotypes. These include four canonical phenotypes of differentiated T cells (Th0, Th1, Th2, and iTreg) as well as six complex phenotypes, whereby multiple lineage-specifying TFs are co-expressed. Our results also suggest that the input dosage regulates the balance of specific T cells within the complex T cell phenotypes, providing new insights into specific patterns of environmental input composition and dosage effects on T cell differentiation.

RESULTS

Mechanistic Logical Model of T Cell Differentiation

A comprehensive mechanistic, logic-based model of T cell differentiation was constructed using regulatory information from published literature. The model includes 38 components and 96 biochemical interactions that regulate the differentiation process of major T cell subtypes, such as Th1, Th2, Th17, and iTreg cells. The individual components of the model represent lineage-specifying TFs (Tbet, GATA3, RORγt, and Foxp3), STAT proteins, cytokines, their receptors, and other signaling molecules. The extracellular environment is represented in the model by eight cytokines and a (generic) TCR ligand, known to play a role in T cell differentiation. The network representation of the model is visualized in **Figure 1**. The regulatory interactions in the model are defined as Boolean functions, which are composed of the "AND," "OR," and "NOT" operators (Supplementary Datasheets 1 and 2). The fully annotated model is available for download in a number of formats [including SBML-qual (Chaouiya et al., 2013)], as well as for viewing, and performing simulations, analyses, and additional modifications within the Cell Collective modeling platform[1] (Helikar et al., 2012b, 2013). The model can be accessed directly at: https://www.cellcollective.org/#6678/cd4-t-cell-differentiation.

The model was validated to ensure that it can reproduce differentiation into four canonical phenotypes (Th1, Th2, Th17, and iTreg), as a result of cytokine stimulation and TCR activation (Supplementary Table 1). Furthermore, the model was able to reproduce more complex behaviors (**Figure 2**). For example, Becskei and Grusby (2007) studied the synergistic effect of positive feedback loops on the expression of the IL-12 receptor (IL-12R). They showed that the number of IFN-γ positive cells and the expression of IL-12R increased when induced by the combination of IL-12 and IL-27. As shown in **Figures 2A,B**, simulations of the presented model under similar experimental conditions resulted in the same qualitative behavior. Furthermore, it has been experimentally shown that the IL-6 regulates the balance between iTreg and Th17 cells in a dose-dependent manner (Yang et al., 2008; Kimura and Kishimoto, 2010). Similarly, simulations of the model show a clear distinction between iTreg and Th17 in an IL-6-dependent manner (**Figure 2C**). Finally, simulations of the model, under environmental conditions similar to those that have been shown to induce the mixed Th1–Th2 behavior (Peine et al., 2013), also resulted in a complex phenotype with activation of both Tbet and GATA3 TFs (**Figure 2D**).

[1] https://www.cellcollective.org

Novel T Cell Phenotypes Are Predicted by Logical Modeling

With the validated model in hand, we sought to understand its capacity to represent various T cell phenotypes. By using ergodic set analysis [see the section "Materials and Methods" and Todd and Helikar (2012)], we explored the state space of the model under 512 possible combinations of the extracellular stimuli (*input compositions*) (**Figure 3A**).

A total of 101 ergodic sets (*outputs*) were obtained as a result of 511 input compositions (Supplementary Table 2). Out of the 511 compositions, 45 input compositions resulted into fixed points (a single remaining input composition was not analyzable even on a supercomputer due to the large size of state space which could not be computed on a feasible temporal scale). The number of input compositions for each output ranged from 1 to 51 (**Figure 3B**). We obtained one output (output 3) that can be stimulated by the maximum of 51 input compositions. Two outputs (outputs 6 and 13) were each stimulated by the maximum of 48 input compositions (**Figure 3B**). Furthermore, four outputs (outputs 10, 22, 29, and 32) were each achieved by 16 different input compositions. All outputs that are individually stimulated by 16 or more input compositions have input compositions with an inactive TCR ligand.

The number of input compositions for the remaining outputs varied from 1 to 4. These input compositions contained an active TCR ligand. In this group, a total of 37 outputs were obtained, whereby each of them was stimulated by four input compositions. A total of 56 outputs were each stimulated with two input compositions. Only one output was stimulated by a single input composition.

Thus, 7 (out of 101) outputs were achieved when stimulated by 211 input compositions with the absence of a TCR ligand. On the other hand, 94 outputs (out of 101) were obtained when stimulated by 255 input compositions with an active TCR ligand. Therefore, fewer outputs (101) have been observed than the total number of input compositions (511), suggesting that a specific cell fate (output) can result from multiple signal compositions, processed by a cell based on biochemical rules in a signaling network (Helikar et al., 2008; Balázsi et al., 2011; Palau-Ortin et al., 2015).

Next, we explored the biological relevance of the produced outputs. As the model centers on the regulation of T cell phenotypes and the TFs related to each differentiated T cell subtype, we classified all the outputs based on the presence of the four TFs (GATA3, Tbet, RORγt, and Foxp3). We found that the model outputs (as a result of the 511 input compositions) cluster into 10 biologically relevant phenotypes. These include the canonical (single cell type) phenotypes as well as the complex phenotypes having more than one lineage-specifying TF.

Specifically, we found four canonical T cell phenotypes that carried Tbet, GATA3, or Foxp3, representing Th1, Th2, and iTreg, respectively (**Figure 3C**). Furthermore, we found that 219 input compositions resulted in nine outputs with no TFs

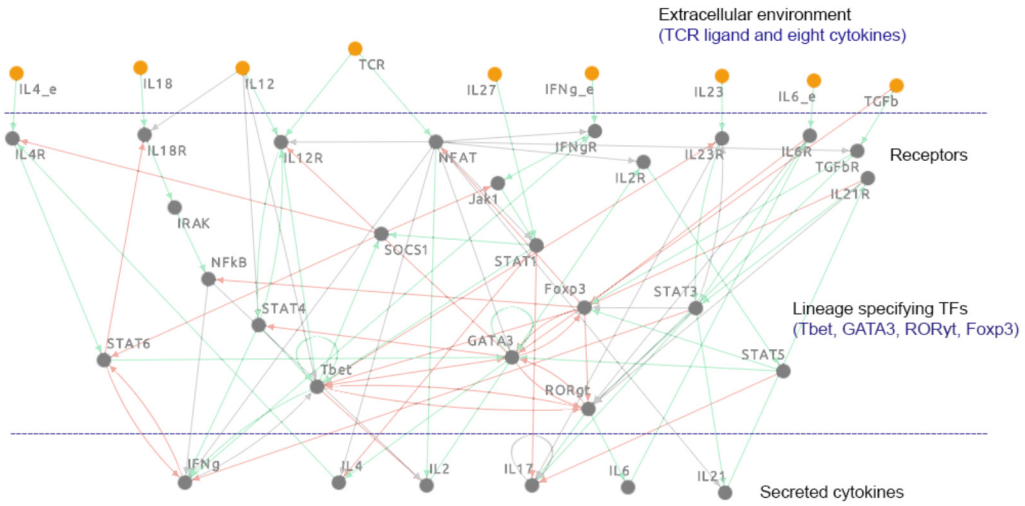


FIGURE 1 | Network diagram of the logical model of signal transduction in CD4$^+$ T cells. The modeled pathways reflect the canonical pathways known to regulate T cell differentiation into the major effector subtypes (Th1, Th2, Th17) and the regulatory subtype (iTreg). The model includes 38 components, including four lineage-specifying TFs (Tbet, GATA3, RORγt, and Foxp3) and nine extracellular components: TCR-ligand, IFN-γ, TGF-β, IL-4, IL-6, IL-12, IL-18, IL-23, and IL-27. Green arrows represent activation, red arrows represent inhibition, and gray arrows represent the conditions associated with activatory or inhibitory interactions. IFNg_e, interferon-γ (external); IL12, interleukin 12; IL18, interleukin 18; IL23, interleukin 23; IL27, interleukin 27; IL4_e, interleukin 4 (external); IL6_e, interleukin 6 (external); TCR, T cell receptor; TGFb, transforming growth factor-β; Foxp3, Forkhead box P3; GATA3, GATA-binding protein 3; IFNg, interferon-γ; IFNgR, interferon-γ receptor (generic); IL12R, interleukin 12 receptor (generic); IL17, interleukin 17; IL18R, interleukin 18 receptor 1; IL2, interleukin 2; IL21, interleukin 21; IL21R, interleukin 21 receptor; IL23R, interleukin 23 receptor; IL2R, interleukin 2 receptor; IL4, interleukin 4; IL4R, interleukin 4 receptor; IL6, interleukin 6; IL6R, interleukin 6 receptor; IRAK, interleukin-1 receptor associated kinase 1; Jak1, Janus kinase 1; NFAT, nuclear factor of activated T cells 5, tonicity-responsive; NF-κB, nuclear factor of kappa light polypeptide gene enhancer in B cells (generic); RORgt, RAR-related orphan receptor C; SOCS1, suppressor of cytokine signaling 1; STAT1, signal transducer and activator 1; STAT3, signal transducer and activator 3; STAT4, signal transducer and activator 4; STAT5, signal transducer and activator 5; STAT6, signal transducer and activator 6; Tbet, T-box expressed in T cells; TGFbR, transforming growth factor-β receptor (generic).

present (Th0 phenotype). Most of the outputs that represent the Th0 phenotype (>95%) were stimulated by the input compositions with an inactive TCR ligand. The remaining Th0-leading input compositions contained an active TCR ligand along with IL-23, or IL-18, or IL-6. This corresponds to the experimentally established scenarios, whereby the T cells cannot differentiate in the absence of TCR activation or in the absence of key lineage-specific cytokines (Podojil and Miller, 2009; Zhu et al., 2010; Chen and Flies, 2013). Fifty-two input compositions led to 16 outputs with active Tbet, representing the Th1 phenotype. A total of 24 input compositions produce 10 outputs with active GATA3, representing the Th2 phenotype, while four input compositions led to one output with active Foxp3, representing the iTreg phenotype. We did not observe distinct outputs with only RORγt active; instead, RORγt was part of the complex phenotypes (discussed below).

In addition to the four canonical phenotypes, the model predicted six complex phenotypes. The number of input compositions, and the number of outputs that represent each complex phenotype, is summarized in **Figure 3C**. Of the six complex phenotypes, three of them including Th1–Th2 (Hegazy et al., 2010; Evans and Jenner, 2013; Peine et al., 2013), Th1–iTreg (Koch et al., 2009), and Th17–iTreg (Eisenstein and Williams, 2009) were experimentally observed earlier, thus further validating the model. The model also predicted three novel complex phenotypes, Th1–Th2–iTreg, Th1–Th17–iTreg, and Th1–Th2–Th17–iTreg, for which experimental validation is foreseeable.

Cytokine Composition Establishes T Cell Phenotypes

Once the classification of all the model outputs into biologically relevant phenotypes was carried out, we analyzed the input compositions (environmental conditions) leading to each of the 10 biological phenotypes. This analysis resulted in 27 patterns of input compositions (**Figure 4**). We also identified the minimal input compositions that are needed to stimulate each phenotype (**Figure 5**). Additionally, the signal transduction sub-networks activated for each phenotype, simulated under a representative input composition, are shown in **Figure 6**.

As indicated in the model validation section, we found that the canonical phenotypes (Th0, Th1, Th2, and iTreg) are regulated by one or more cytokines. We also found that all the complex phenotypes can be stimulated by more than one input composition. Strikingly, our modeling effort predicts that in order to induce specific phenotypes, certain cytokines cannot be co-present in a given input composition (**Figures 4**, **5**). For example, based on our model, TGF-β should not be present in the input compositions leading to the Th1–Th2 phenotypes, and IL-6 should be absent from the input compositions that lead to iTreg, Th1–iTreg, and Th1 phenotypes. On the other hand, IL-4 can be present in the input composition leading to

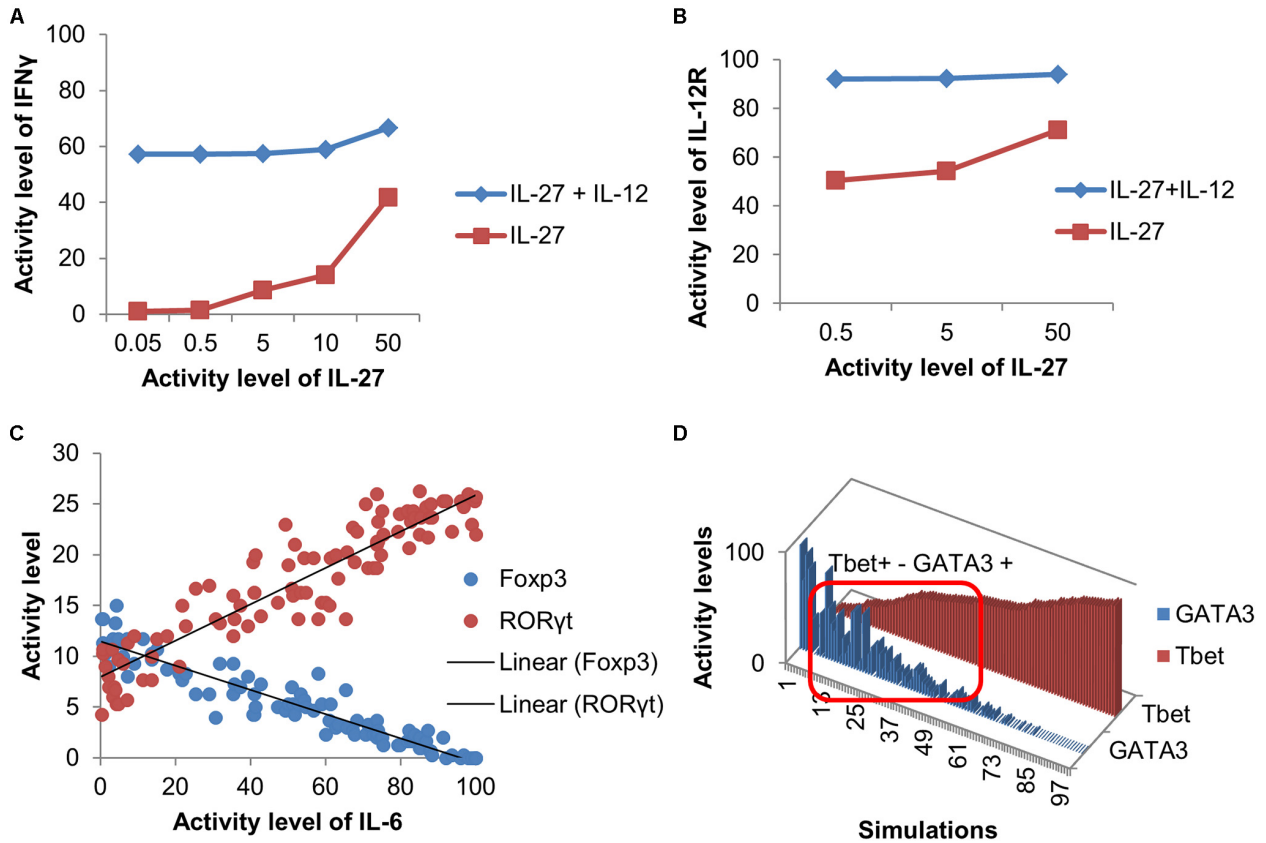
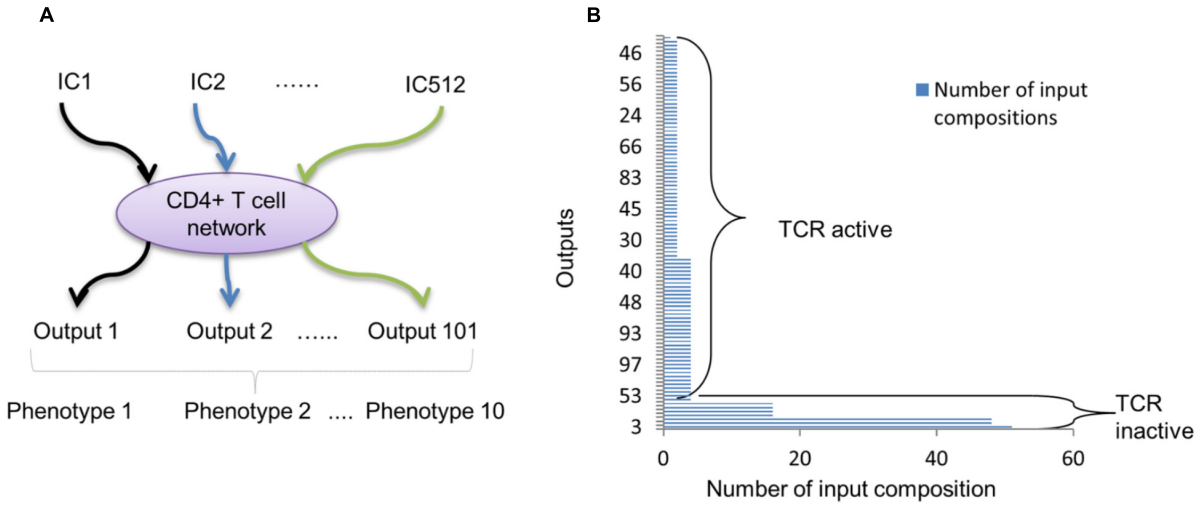


FIGURE 2 | Validation of the model using complex behaviors. **(A–B)** Simulations of the model under experimental conditions using the same concentration ratio of IL-27 and IL-12 (i.e., 1000, 100, 10, 1 for IFN-γ and 100, 10, 1 for IL-12R) as in Becskei and Grusby (2007) show that combination of IL-27 with IL-12 leads to a synergistic effect on level of IFN-γ production and the activity of the IL-12 receptor (IL-12R). **(C)** Simulated IL-6 dose-response effect on the activation of Foxp3 and RORγt. **(D)** Model simulations reproduce a mixed Th1/Th2 phenotype with varying levels of Tbet and GATA3. The significant activity levels of both Tbet and GATA3 are observed in the area inside the red box.

Th1, but only when co-present with IL-6. IL-4 also needs to be absent in input compositions leading to iTreg, Th17–iTreg, Th1–iTreg, and Th1–Th17–iTreg phenotypes. Finally, IL-12 and IL-18 cannot be co-present in the extracellular environment that stimulates differentiation into Th1, Th2, Th1–iTreg, Th1–Th17–iTreg, and Th0 (in the absence of the TCR ligand) phenotypes.

The previously mentioned heterogeneous and conditional effect of combining IL-12 and IL-18 is also supported and partially explained through experimentally described regulatory mechanisms (Yoshimoto et al., 1998; Nakanishi et al., 2001). Specifically, we observed that combining IL-18 with IL-12 favors co-expression of Tbet, GATA3, and Foxp3. It was previously shown that combining IL-12 and IL-18 can synergistically increase the Tbet-stimulated IFN-γ production in Th1 cells (Tominaga et al., 2000). In another study, it was shown that IL-18, but not IL-12, increases the production of IFN-γ by CD8+ and $CD4^+$ T cells in the K14E7 transgenic skin (Gosmann et al., 2014). Further, the combination of IL-12 and IL-18 has been shown to induce the production of IFN-γ in the absence of antigen (Munk et al., 2011). Finally, it has been shown that IL-18 in the absence of IL-12 can stimulate Th2 response (Nakanishi et al., 2001).

To further investigate the effect of IL-12 and IL-18 on the Th1–Th2–iTreg phenotype, the model was simulated under the input composition of IL-12, IL-18, and TCR (with all other cytokines inactive). Simulation results suggested the synergistic effect of IL-12 and IL-18 on the activity level of GATA3 and Foxp3. Interestingly, the increased activity of GATA3 and Foxp3 was observed in the absence of external IL-4 and TGF-β (**Figure 7A**), suggesting that the combination of IL-12 and IL-18 (while controlling for the TCR signal) are able to stimulate the Th1–Th2–iTreg phenotype in an IL-4- and TGF-β-independent manner. We also found that the combination of IL-12 and IL-18 is a weaker activator of GATA3 and Foxp3 (**Figure 7A**). This is because the IL-12 can also stimulate Tbet, which in turn suppresses the GATA3 and Foxp3. Results obtained from the simulated IL-12R knock-out suggested an eightfold increase in the activity of GATA3, whereas the overexpression of IL-12R slightly decreased the activity levels of GATA3 and Tbet. Knock-out of IL-18R resulted in a complete inactivation of GATA3 and Foxp3,



C

No. of input compositions	No. of outputs	Presence/ Absence of lineage specifying TFs				Phenotype
		Foxp3	GATA3	RORγt	Tbet	
52	16					Th1
24	10					Th2
4	1					iTreg
20	6					Th1-iTreg
40	16					Th1-Th2
4	1					Th17-iTreg
50	21					Th1-Th2-iTreg
20	7					Th1-Th17-iTreg
33	14					Th1-Th2-Th17-iTreg
219	9					Th0

FIGURE 3 | Simulation of T cell model under all possible environmental conditions. **(A)** A schematic diagram showing simulation strategy using all possible input compositions (IC). **(B)** A total of 101 outputs (ergodic sets) were obtained. The number of input compositions stimulating each ergodic set is ranging from lowest 1 to highest 51. Only one ergodic set resulted from a single input composition. **(C)** Ten phenotypes based on presence and absence of lineage-specifying TFs were obtained. The number of input compositions and ergodic sets for each phenotype are provided in the first and second column. Based on the presence and absence of TFs, each phenotype was determined. Blue cells correspond to TFs present in the phenotype, whereas the orange cells represent inactive TFs.

whereas the overexpression of IL-18R resulted in a greater than twofold increase in the activity levels of GATA3 and Foxp3. These results indicate that the knock-out of IL-12R favors Th2 phenotype, whereas the knock-out of IL-18R favors Th1 phenotype under Th1–Th2–iTreg stimulating environmental conditions.

The differentiation to Th2 was previously observed in airway epithelia in the presence of IL-18, but not IL-4 (Murai et al., 2012). The IL-4-independent Th2 stimulation possibly occurs through the STAT5-mediated GATA3 activation (Yamane et al., 2005; Paul, 2010). The IL-18R1 signaling was also found to promote Foxp3+ iTreg cell function within colonic lamina propria (Harrison et al., 2015). To better understand the mechanism of how the IL-18 and IL-12 can stimulate GATA3 and Foxp3, we further analyzed the network structure of the model. We found that IL-12 and IL-18 can possibly induce the production of IL-2, which stimulates GATA3 and Foxp3 in STAT5-dependent pathways (**Figure 7B**). The knock-out simulation of NF-κB or STAT5 resulted in complete inactivation of GATA3 and Foxp3. On the other hand, the overexpression of STAT5 increased the mean activity level of Foxp3 by 62-fold, while no change in activity levels of GATA3 was observed. The simulated over-expression of NF-κB had shown 5.4-fold and twofold increase in the activity levels of Foxp3 and GATA3, respectively. These results predict the role of IL-12 and IL-18 in stimulation of the Th1–Th2–iTreg phenotype in an NF-κB- and STAT5-dependent manner (**Figure 7C**). Furthermore, our simulation results suggest that a combination of IL-18 and IL-12 can stimulate Tbet, GATA3, and Foxp3; however, the activity levels of GATA3 and Foxp3 were lower than that of Tbet (**Figure 7A**). Additionally, we have found that IL-12 and IL-18 combination in the presence of IL-6 can stimulate the

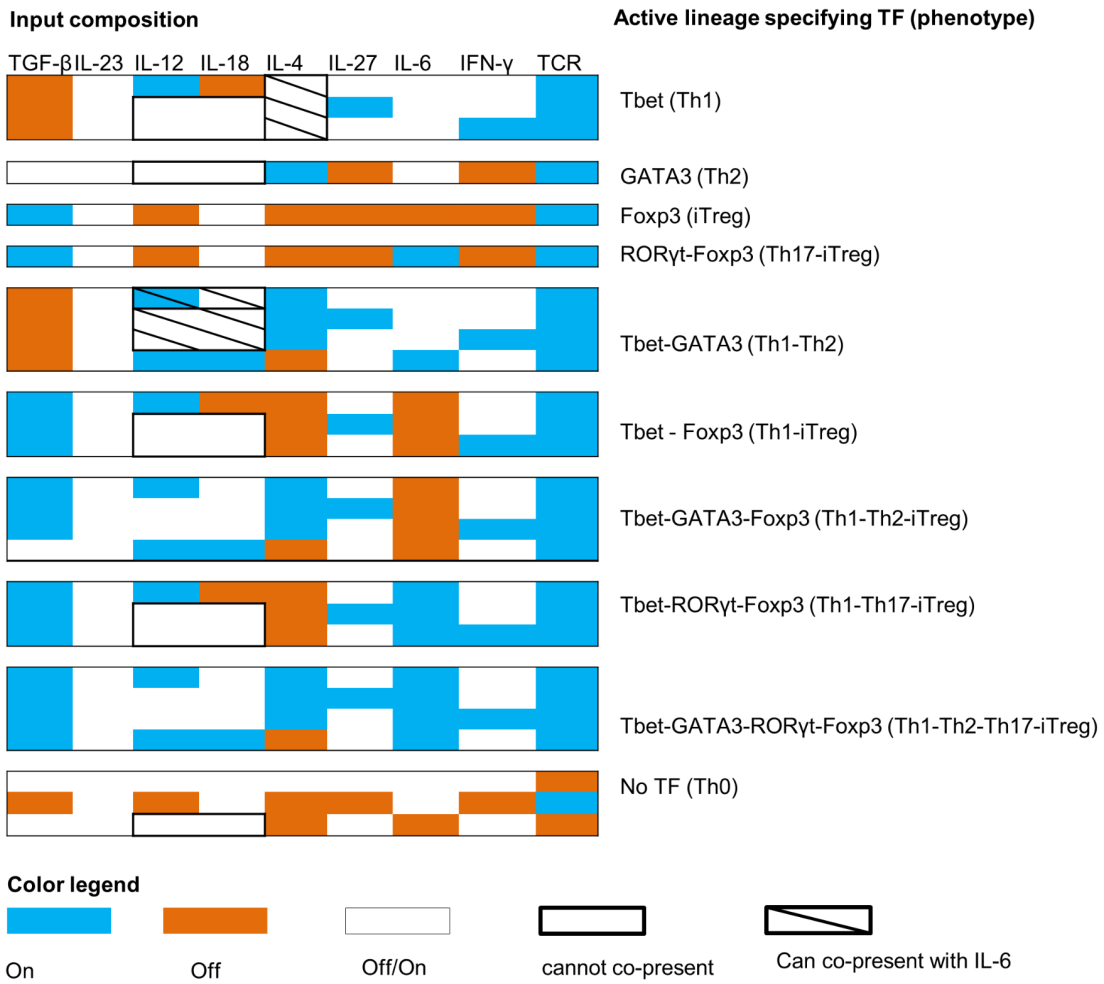


FIGURE 4 | Input compositions for all T cell phenotypes. The color map shows the patterns of input compositions that give rise to each observed phenotype. For example, co-expression of RORγt–Foxp3 needs TCR + IL-6 + TGF-β (+IL-18 and IL-23 can also be active) to be active and IL-12, IFN-γ, IL-27, and IL-4 to be inactive.

Th1–Th2 phenotype (Supplementary Figure 1 in Supplementary Datasheet 3).

Altogether, we have identified input composition patterns that include the minimum combinations of cytokines required to stimulate a particular T cell phenotype, as well as complete pattern of cytokines that can be co-present to stimulate a given phenotype (Supplementary Table 3). Our results also predict the relevance of IL-12 and IL-18 in regulating the Th1–Th2–iTreg phenotype. Finally, we predicted an alternative pathway that can stimulate GATA3 and Foxp3 in an IL-4 and TGF-β-independent manner.

Cytokine Dosage Determines the Balance Between Complex T Cell Phenotypes

In the previous section, various input compositions that lead to different canonical and complex phenotypes were characterized. The logical question that we raise now is: How is the balance of each T cell subtype within a complex phenotype controlled?

As indicated in the "Introduction" section, several reports suggest that the balance between Th17 and iTreg is regulated by the dosage of IL-6 (Kimura and Kishimoto, 2010; Omenetti and Pizarro, 2015). To explore how the input dosages within each composition affect the complex phenotypes, we analyzed the model under various activity levels of cytokines and the TCR ligand under the complete set of input compositions.

We used the representative input compositions for each identified phenotype as described in **Figures 4**, **5**. Specifically, we used two types of representative input compositions from each row in **Figure 4**. The two types include, one with the maximum number of inputs that can be simultaneously present to stimulate a specific T cell phenotype, and a second type that is represented by input compositions consisting of the minimal number of inputs required to stimulate the identified phenotypes.

Sensitivity analysis of the model was performed to describe the effect of each input in its composition on the TF(s) for

FIGURE 5 | Minimal input compositions required to stimulate T cell phenotypes. All inputs in blue boxes are required to stimulate the corresponding phenotype. For example, the Th1–Th2 phenotype can be stimulated by input composition: TCR ligand + (IFN-γ OR IL-12 OR IL-27) + (IL-4 OR IL-12 + IL-18 + IL-6). Minimum three inputs are required to stimulate the Th1–Th2 phenotype (e.g., TCR ligand + IFN-γ/IL-12/IL-27 + IL-4).

the corresponding complex phenotype (**Figure 4**). The analysis predicted individual inputs that are important for regulating the balance among lineage-specifying TFs. For example, for the Th1–Th2 phenotype, when stimulated with a maximum of eight inputs, the sensitivity analysis suggested that IL-27, IFN-γ, and IL-12 are negatively correlated with GATA3 (**Figures 8A–C**). The TCR signal is negatively correlated with GATA3 [partial correlation coefficient (PCC) range = −0.18 to −0.19] under three input compositions (**Figures 8A–C**). Interestingly, a positive correlation between the TCR ligand and GATA3 was observed when the Th1–Th2 phenotype was stimulated in the absence of IL-4 (and in the presence of IL-12, IL-18, and IL-6) (**Figure 8D**). On the other hand, the IL-18 had a moderate negative correlation with Tbet (PCC range = −0.28 to −0.29) under all tested input compositions (**Figures 8A–D**). The IL-4 had a very low correlation with Tbet (PCC range = 0.005–0.01) under all tested input compositions (**Figures 8A–D**).

Next, in the case of the Th1–Th2 phenotype stimulated under minimal input compositions, higher correlations between the inputs and TFs were observed compared to the correlations observed with maximal input compositions (**Figures 8E–H**). In the case of the input composition "TCR + IL-12 + IL-4," IL-12 had a strong negative correlation (PCC = −0.68) with GATA3, and a strong positive correlation with Tbet (PCC = 0.7). In this case, the TCR ligand had a moderate negative and

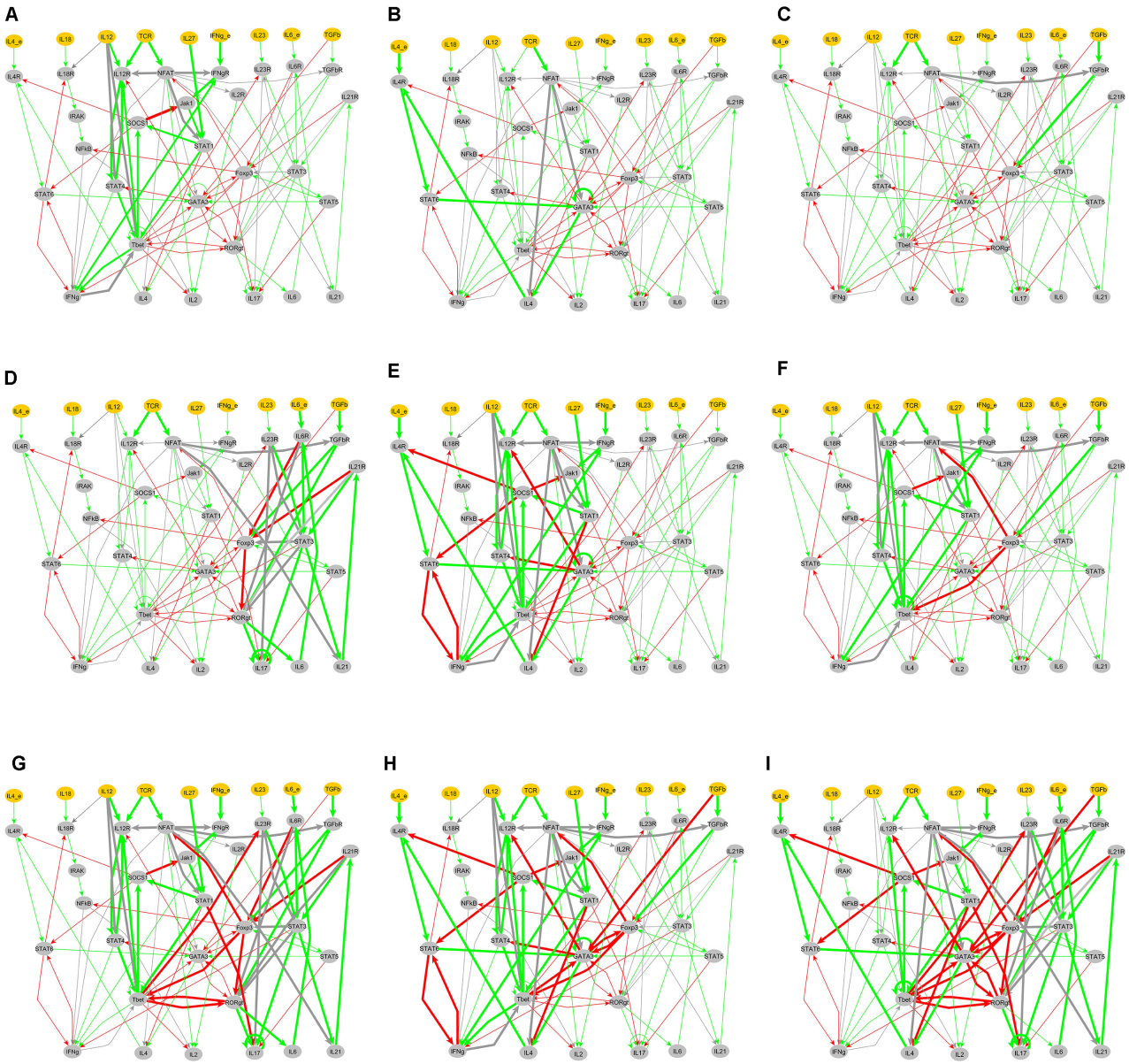


FIGURE 6 | Active signal transduction sub-networks for T cell phenotypes. The active sub-networks under an input composition are mapped on the entire network of T cell differentiation and are marked by bold arrows. Green arrows represent activation and red arrows represents inhibition, and gray arrows represent the condition associated with activation and inhibition. **(A)** Th1 phenotype under input composition: TCR-ligand + IL-12 + IL-27 + IFN-γ, **(B)** Th2 phenotype under input composition: TCR-ligand + IL-4, **(C)** iTreg phenotype under input composition: TCR + TGF-β, **(D)** Th17-iTreg phenotype under input composition: TCR-ligand + IL-6 + TGF-β, **(E)** Th1–Th2 phenotype under input composition: TCR-ligand + IL-4 + IL-12 + IL-27 + IFN-γ, **(F)** Th1–iTreg phenotype stimulated under input composition: TCR-ligand + IL-12 + IL-27 + IFN-γ + TGF-β, **(G)** Th1–Th17–iTreg phenotype stimulated under input composition: TCR-ligand + IL-12 + IL-27 + IFN-γ + IL-6 + TGF-β, **(H)** Th1–Th2–iTreg phenotype stimulated under input composition: TCR-ligand + IL-4 + IL-12 + IL-27 + IFN-γ + TGF-β, **(I)** Th1–Th2–Th17–iTreg phenotype stimulated under input composition: TCR-ligand + IL-4 + IFN-γ + IL-6 + TGF-β.

positive correlation with GATA3 (PCC = −0.25) and Tbet (PCC = 0.22), respectively. In the case when Th1–Th2 phenotype was stimulated under input composition "TCR + IL-12 + IL-18 + IL-6," the TCR ligand was positively correlated with both GATA3 (PCC = 0.30) and Tbet (PCC = 0.30). A strong positive correlation was observed between IL-4 and GATA3 (PCC = 0.65) under the input composition "TCR + IFN-γ + IL-4." In the Th1–Th2 complex phenotype, we observed that the TCR ligand is negatively correlated with GATA3. The negative effect of a strong TCR ligand signal on GATA3 is in agreement with the earlier studies suggesting that a strong TCR signal can promote a strong Th1 response, whereas a weaker signal favors the Th2 response

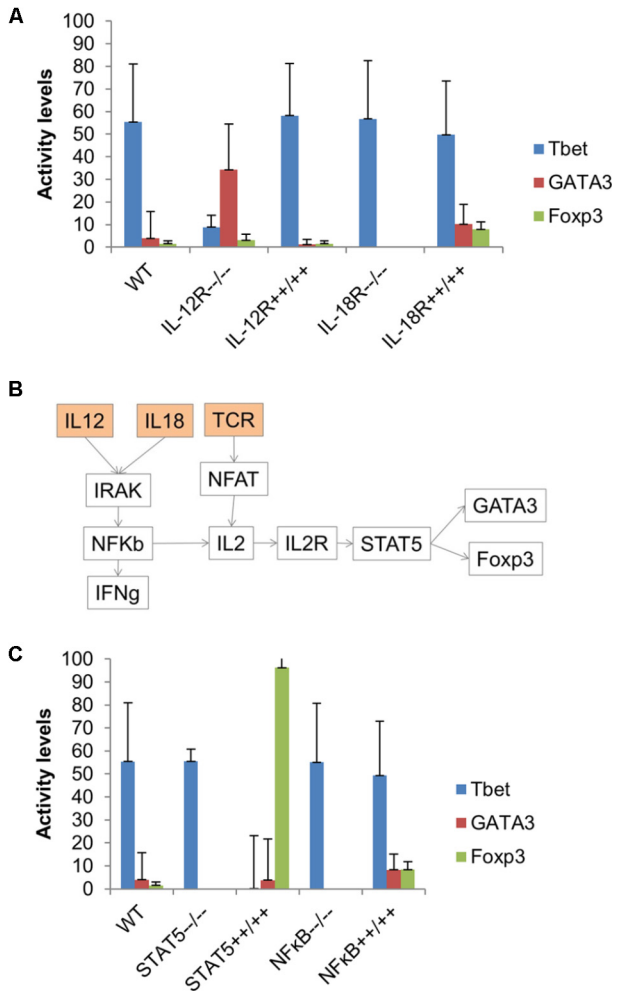


FIGURE 7 | Combination of IL-12 and IL-18 favors the Th1–Th2–iTreg phenotype. **(A)** The model was simulated using input composition (TCR ligand + IL-12 + IL-18) under five conditions, including wild-type, and knock-out and overexpression of both IL-12R and IL-18R. Overexpression of IL-12R favors Th1 phenotype, whereas IL-12R knock-out significantly increased the activity level of GATA3. IL-18R knock-out completely inactivated both GATA3 and Foxp3. **(B)** A subnetwork that might induce GATA3 and Foxp3 in the presence of IL-12 and IL-18, but in the absence of IL-4 and TGF-β. **(C)** Knock-out of STAT5 and NF-κB resulted in complete inactivation of GATA3 and Foxp3, whereas overexpression of STAT5 induced a strong Foxp3 response. Overexpression of NF-κB increased the activity of both GATA3 and Foxp3 by more than twofold.

(van Panhuys et al., 2014). The sensitivity analysis results for all other mixed phenotypes are provided in Supplementary Table 4.

In summary, the sensitivity analysis of our model predicts "driver" inputs. Furthermore, it characterizes the strength and direction (positive or negative) of the effect inputs can have on the regulation of the balance of each T cell subtype within the complex phenotypes. The strength of association between the inputs and TFs varied based on the number of inputs in the input compositions.

Determining the Optimal Input Dosage Regulating the Balance Between Complex Phenotypes

In the previous sections, the predicted complex T cell phenotypes, input compositions, as well as the potential dosage effect each input can have on the phenotype, were discussed. Next, we examined the specific activity levels of the input compositions required to control each specific T cell phenotype. The model was simulated under 10,000 randomly generated environmental conditions within the context of each relevant input composition. Results from these simulations provided us with specific input activity levels that have a low coefficient of variance (CV) in activity levels of co-expressed lineage-specifying TFs. Specifically, we investigated and characterized the activity levels for each input composition that will drive a complex T cell phenotype to each of the T cell subtypes or a balanced mixed phenotype by maximizing the activity levels of the respective TFs. For example, to achieve a balanced Th1–Th2 phenotype that has similar activity levels to that of GATA3 and Tbet, we characterized the optimal activity levels for each input in the Th1–Th2-leading input compositions. This predicted optimal input composition includes the low activity of the TCR ligand, IFN-γ, IL-12, and IL-27, medium activity of IL-18 and IL-6, and high activity of IL-4. The activity of IL-23 can vary from low to high, whereas TGF-β should be inactive (**Figure 9A**).

To illustrate the effect of using optimal activity levels, we stimulated the Th1–Th2 phenotype by using a median value of optimal activity level for each input. As expected, the simulation results show similar activity levels of Tbet and GATA3 (**Figure 9B**). To further investigate the effect of dominant inputs (identified from the sensitivity analysis) on the Tbet–GATA3 combination, we performed dose-response analysis by varying the dominant cytokines while fixing the other inputs to median activity levels (**Figure 9A**). As expected, our results (**Figures 9C–J**) suggest that the increased signal strength of TCR ligand or increased activity of IL-12 and IL-27 can drive the Th1–Th2 phenotype toward Th1 by increasing the activity of Tbet and decreasing the activity of GATA3. In contrast, the increased activity of IL-18 can drive Th1–Th2 phenotypes toward Th2.

DISCUSSION

In this study, we sought to investigate the cellular phenotypes as a result of $CD4^+$ T cell differentiation under diverse environmental conditions and understand how the balance between complex phenotypes is regulated. To achieve this, by manually curating literature data, we constructed a mechanistic computational (logical) model of signal transduction that regulates the differentiation of naive T cells into Th1, Th2, Th17, and iTreg cells. The components (i.e., proteins and genes) in a logical model can have binary (0 or 1) states at any time *t*. The state of the network evolves stepwise based on the logical rules defined for each model component (Helikar and Rogers, 2009; Helikar et al., 2012a,b, 2013; Naldi et al., 2015; Abou-Jaoudé et al., 2016; Barberis and Verbruggen, 2017; Linke et al., 2017).

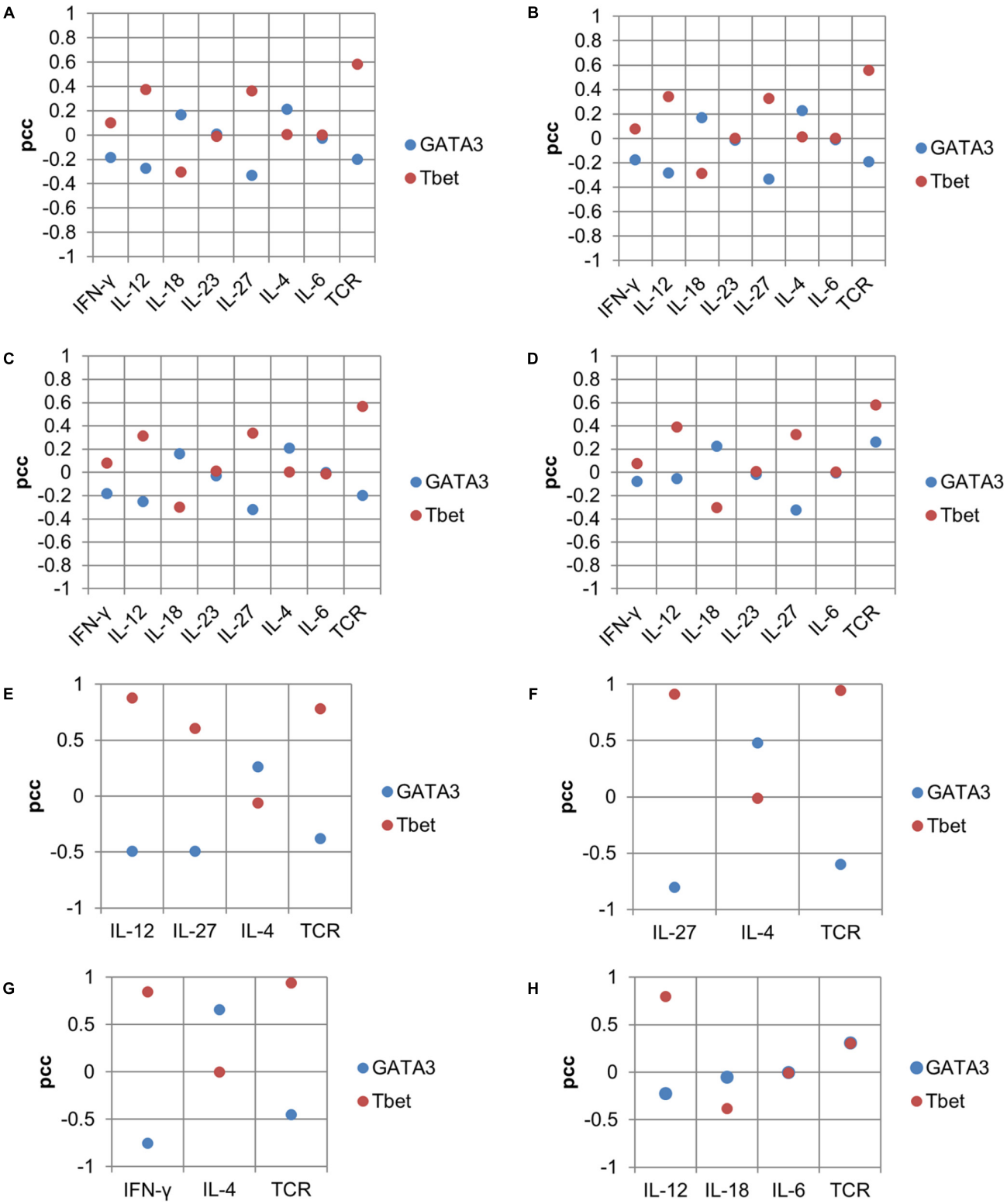


FIGURE 8 | Sensitivity analysis showing the input effect on lineage-specifying TFs for the Th1–Th2 phenotype. Panels **(A–D)** are based on simulations using maximal input compositions. Panels **(E–H)** are based on minimal input compositions. PCC as a measure of association between inputs (cytokines and TCR) and lineage-specifying TFs is shown on *Y*-axes and input composition (cytokines and TCR) is shown on *X*-axes.

A

Co-expressed TFs	Optimal settings of input combinations								
	IFN-γ	IL-12	IL-18	IL-23	IL-27	IL-4	IL-6	TCR	TGF-β
Tbet-GATA3	11.20	16.10	58	38.30	18	83.50	46.60	13.50	X
Foxp3-RORγt	X	X	48.50	33.80	X	X	6.4	65.70	97.70
Tbet-Foxp3	62	99	X	45	40	X	X	97	96
	51.4	X	90.4	X	99.8	X	X	86.8	83.4
Tbet-Foxp3-RORγt	39.30	19.80	X	46.80	31.60	X	6.30	72.60	95.60
	6.40	X	53.5	74.30	24.9	X	24.6	33.30	91
Tbet-GATA3-Foxp3	16.80	14.60	45.7	86.90	10.10	81.50	X	59.1	64.60
Tbet-GATA3-RORγt-Foxp3	13.4	10.2	58.5	43.6	8.4	56	18.7	39.9	81.1

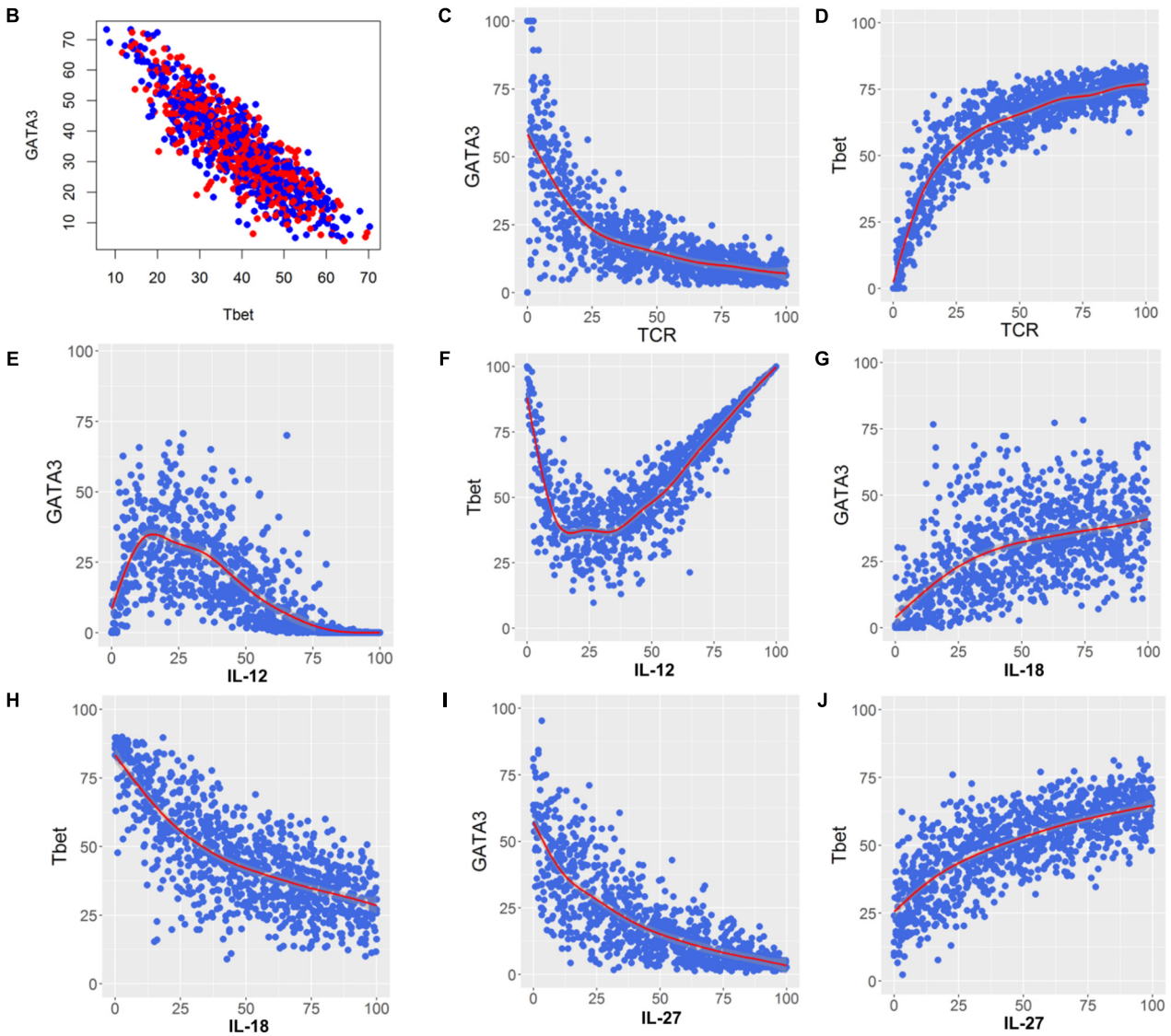


FIGURE 9 | Dose-dependent regulation of complex phenotypes co-expressing lineage-specifying TFs. **(A)** Optimal settings using maximal input composition that stimulate co-expression of lineage-specifying TFs. Shown are median activity levels of inputs that result in a balanced activity level of both TFs. Activity levels ranging from 0 to 40 (blue) were considered as low, 40 to 80 were medium (light blue), and 80 to 100 were high (purple). **(B)** Activity levels of GATA3 and Tbet when simulated under optimal input settings that stimulate the Th1–Th2 phenotype. Median activity levels from panel A were used for the simulation. **(C–J)** Dose-response curves illustrating the effect of TCR, IL-12, IL-18, and IL-27 on GATA3 and Tbet.

We systematically characterized the model's dynamics in the context of activity of lineage-specifying TFs under 511 input compositions consisting of eight cytokines and a TCR signal. In addition to the dynamics representing the classical Th0, Th1, and Th2 phenotypes, we found several complex (mixed) phenotypes (dynamics with more than one lineage-specific TFs), including Th1–Th2, Th1–iTreg, Th17–iTreg, Th1–Th2–iTreg, Th1–Th17–iTreg, and Th1–Th2–Th17–iTreg. Our results are in agreement with recent studies that reported hybrid T cell phenotypes *in vitro* and *in vivo* (Zhou et al., 2008; Peine et al., 2013). Stable complex Th1–Th2 phenotypes parallel to the classical Th2 phenotypes were observed *in vivo* upon infection mediated by parasites *Schistosoma mansoni* and *Heligmosomoides polygyrus* (Peine et al., 2013), as well as by the threadworm *Strongyloides stercoralis* (Bock et al., 2017). Moreover, Th1–iTreg intermediate phenotypes were observed during Th1 polarizing infections (Koch et al., 2009; Oldenhove et al., 2009; Evans and Jenner, 2013). In a recent system level study, a continuum of T cell differentiation states with stable co-expressed lineage-specific TFs has been observed when stimulated under different combinations of six cytokines (Eizenberg-Magar et al., 2017).

Interestingly, we did not observe a canonical Th17 (RORγt-only) phenotype. Instead, our model predicts the existence of a mixed Th17–iTreg phenotype. This result can be partially explained by the fact that both Th17 and iTreg share a common mechanism by cytokine TGF-β, and the differentiation of naive T cells into iTreg or Th17 depends on the cytokine-driven (TGF-β and IL-6) balance of lineage-specifying TFs Foxp3 and RORγt (Omenetti and Pizarro, 2015). In addition, it is known that the Th17/Treg balance is critical to maintain immune tolerance. The imbalance of Th17/Treg has been observed in the peripheral blood of cervical cancer patients (Chen et al., 2013), non-small cell lung cancer patients (Duan et al., 2015), and in patients with chronic low back pain (Luchting et al., 2014). Thus, the complex Th17–iTreg phenotype might play an important role in maintaining Th17/Treg homeostasis. Such complex RORγt–Foxp3 co-expressing T cells were observed in an autoimmune diabetes model (Ichiyama et al., 2008; Tartar et al., 2010), in the lamina propria (Zhou et al., 2008), in the peripheral blood and tonsils (Voo et al., 2009), and in the large intestine (Ohnmacht et al., 2015; Fang and Zhu, 2017). It is also possible that the lack of Th17-only phenotype is due to the incomplete nature of the model. However, it suggests that additional experimental validation may be required to better understand the relationship and mechanism of switching between iTreg and Th17 phenotypes.

We also predicted novel phenotypes that have the potential to have three active TFs (Tbet–GATA3–Foxp3, Tbet–RORγt–Foxp3), as well as one with all four TFs (Tbet–GATA3–RORγt–Foxp3). In partial support of our prediction, basal levels of Tbet and GATA3 have been observed in iTreg cells (Yu et al., 2015). While not yet shown experimentally, the Th1–Th17–iTreg phenotype was also predicted by a similar modeling approach (Naldi et al., 2010).

By analyzing all possible inputs combinations, we obtained the minimal and maximal input compositions for each identified phenotype. The minimal composition includes a minimum number of inputs that can stimulate a phenotype. On the other hand, the maximal composition includes a maximum number of inputs that can be simultaneously active to result in the same phenotype. In this analysis, we found that in order to stimulate Th1, Th2, Th1–iTreg, Th1–iTreg, Th1–Th17–iTreg, and Th0 phenotypes, IL-12 and IL-18 cannot be combined in the environment. We observed that the combination of IL-12 and IL-18 leads to the stimulation of GATA3 and Foxp3 even in the absence of IL-4 and TGF-β via a NF-κB-dependent pathway. We predicted that a combination of IL-18 and IL-12 could result in a Th1–Th2–iTreg complex phenotype. Analysis of the model's network structure suggests a potential mechanism that is dependent on NF-κB and STAT5 (**Figure 7B**). Previous studies suggest that IL-18 has a context-specific functional heterogeneity and can induce both Th1 and Th2 T cell phenotypes. The combination of IL-12 and IL-18 has been shown to have a synergistic effect on IFN-γ production that stimulates the Th1 phenotype (Tominaga et al., 2000; Munk et al., 2011). It has also been found that IL-18 alone (without IL-12) can stimulate the Th2 phenotype (Nakanishi et al., 2001). In a study on airway epithelial cells in response to *Alternaria*, it was found that secreted IL-18 has the capacity to stimulate the Th2 phenotype (Murai et al., 2012). Since IL-12 can up-regulate IL-18R expression, it might be possible that the combination of IL-12 and IL-18 may regulate the Th1, Th2, Th1–Th2, and Th1–Th2–iTreg phenotypes in a dose-dependent manner.

Next, the sensitivity analysis of the model suggested that the dosage of the individual inputs regulates the balance within the different complex T cell phenotypes. We investigated the dosage effect by using both minimum and maximum number of inputs under varying activity levels. For example, our results suggest that the dynamics of the complex Th1–Th2 phenotype depend on the combination and dosage of IFN-γ, IL-12, IL-27, IL-18, IL-4, and the TCR ligand. The increased activity levels of the cytokines IFN-γ, IL-12, IL-27, and TCR ligand drive the phenotype toward Th1, whereas the IL-18 or IL-4 drive the Th2 phenotype. The IL-23 and IL-6 have no correlation with either Tbet or GATA3. Under both maximal and minimal input compositions, the IL-4 had low to no correlation with Tbet. On the other hand, the IL-18 was positively correlated with GATA3 and negatively correlated with Tbet. Thus, we predicted that IL-18 may have a dominant role over IL-4 to favor Th2 phenotype under the Th1–Th2 stimulating environmental conditions.

Next, we identified the activity levels of the inputs required to regulate the complex T cell phenotypes. Our results suggest a range of activity levels required to obtain a specific phenotype under minimal and maximal input compositions. For example, a high amount of IL-4 or IL-18 and a low amount of IFN-γ, IL-12, IL-27, and TCR ligand are required to stimulate the Th1–Th2 phenotype under maximal input composition. Low activity of GATA3 under higher TCR ligand activity is indeed in agreement with the literature where it has been shown that a strong TCR signal represses GATA3 (Aguado et al., 2002; Yamane et al., 2005; Paul, 2010; Altin et al., 2011; Yamane and Paul,

2012). Interestingly, our results showed an increase as well as a decrease in the activity levels of GATA3 depending on the activity levels of IL-12. This can be achieved as a result of IL-12 up-regulating IL-18R, which induces NF-κB-mediated GATA3 activation. On the other hand, a higher activity of IL-12 results in a strong Tbet activation, which in turn suppresses GATA3. Although the predicted activity levels are dimensionless and semi-quantitative, they provide a starting point for calibrations against ligand concentrations in specific experimental research protocols.

In summary, results provided in this study can provide a platform to generate and design testable hypotheses in the context of T cell differentiation in response to various combinations and dosage of environmental signals. Furthermore, the presented results and the mechanistic model can be used as tools to further investigate the specific pathway mechanisms that govern each complex phenotype. Input availability and relative dosage at which inputs generate a productive signaling cascade necessarily result in a variable timing of an immune response. Specifically, we and others propose that dosage- and timing-dependent impact of inputs, such as ILs, may impact the T cell differentiation (Barberis et al., 2018; Martinez-Sanchez et al., 2018). This may be investigated by employing experimental methodologies that we have recently envisioned (Barberis and Verbruggen, 2017). Furthermore, crosstalk between ILs and signaling cascades, such as the one governing the cell cycle, may impinge on a timely T cell-mediated protective response (Barberis et al., 2018). These aspects are the focus of our current research efforts. Together with new model-based predictions, improving the understanding of the detailed mechanisms underlying T cell differentiation, can be helpful to design strategies for immunotherapy against pathogens and various diseases of the immune system.

MATERIALS AND METHODS

Model Construction

The computational model is a mechanistic, logic-based model of signal transduction processes known to regulate $CD4^+$ T cell differentiation into Th1, Th2, Th17, and iTreg cells. Each component of the model can assume an active (1) or inactive (0) state at any time *t*. The activity state of the model's internal components is determined by the regulatory mechanisms of other directly interacting components. These regulatory mechanisms are described with Boolean functions (Samaga and Klamt, 2013; Albert and Thakar, 2014; Le Novère, 2015; Naldi et al., 2015; Abou-Jaoudé et al., 2016; Linke et al., 2017).

The new signal transduction model was constructed manually by curating published regulatory mechanisms of each signal transduction component. Each of the 38 components in the model corresponds to a signaling molecule (mainly proteins). The model also contains nine external components that represent the extracellular environment, consisting of eight cytokines (IFN-γ, TGF-β, IL-4, IL-6, IL-12, IL-18, IL-23, and IL-27) and a generic TCR ligand. The final model consists of 38 components (29 internal and 9 external) connected with 96 interactions. The model is fully annotated with published evidence for each component and interaction to ensure transparency and reproducibility. The model is available via the web-based modeling and analysis platform Cell Collective (Helikar et al., 2012b, 2013), accessible at https://www.cellcollective.org (under Published Models) where it can be simulated as well as downloaded (and other logical models published by the community) in several file formats (such as SBML-qual, text file of logical functions, and truth tables).

State Space Analysis

The logical model herein is a *Probabilistic Boolean Control Network (PBCN)* (Todd and Helikar, 2012), whereby each external input (components that are not regulated by other model components) is activated by a user-defined probability of activation (ranging from 0 to 100%). The activity levels of the external inputs and the logical rules associated with each internal node allow the system to update stochastically in time. As such, a *PBCN* is a reducible Markov chain (Tijms, 2003). We used ergodic sets (recurrent communicating classes of the corresponding Markov chain) as a model of stable cell states that represent the phenotype of a differentiated T cell. Ergodic sets are a collection of states in state space such that once the system evolves to one of these states it will remain in this set of states. In this way, the ergodic sets are the stochastic equivalents to attractors in purely Boolean networks (Ribeiro and Kauffman, 2007).

From each initial condition, the system will arrive in one of a (possibly) different collection of ergodic sets. In order to find all the ergodic sets, one would need to let the system evolve from every possible initial condition. Given the large number of possible initial conditions (2^{29}), this is computationally infeasible. Thus, we found those ergodic sets that can be reached from the initial state where all internal components are inactive. This represents our goal, i.e., to identify cell phenotypes that are the result of differentiation from naive T cells (i.e., all model components are inactive). Once an ergodic set was identified it was treated as an *irreducible* Markov chain and thus has an associated limiting distribution. Activities of the internal components are interpreted by approximating the limiting distribution of the Markov chain via simulations in Cell Collective. This means that each internal component has a unit less activity level corresponding to the probability that it is active in the limiting distribution of the Markov chain.

Identification of Ergodic Sets

The extracellular environment (external input components) in the presented model consists of nine stimuli — eight cytokines and a generic TCR ligand. A given extracellular environment is described according to those stimuli that are *off* (no activity) and those that are *on* (some level of activity). Thus, there are $2^9 = 512$ possible *off*/*on* configurations for the extracellular environment (input compositions). The ergodic sets that are reachable from the naive state (where all components are inactive) depend only on this *off*/*on* description and not on the activity level of the non-off cytokines. We were able to identify the corresponding reachable ergodic sets for 508 of these input compositions. The

only extracellular environments that are yet unknown are the three where all stimuli are *on* except for TGF-β, or IL-23, or IL-4. The ergodic sets were identified in two steps. In the first step, Tarjan's algorithm (Tarjan, 1972) was used to identify communicating classes of states. In the second step, these classes were directly tested to determine if they were closed. The ergodic sets (other than the fixed points) ranged in size from the smallest, with two states, to the largest with 594,962 states. These ergodic sets correspond to the "outputs" in **Figures 3A,B**. Each state in an ergodic set specifies the state of the internal network. In order to classify an ergodic set, for each internal component we computed the percentage of states in which the component was active. For example, the ergodic set that was identified when the TCR ligand and IL-4 are *off* while all other external stimuli are *on*, was found to have 64 states. Each of IL-18R, IL-4R, IRAK, NF-κB, and STAT6 are *on* in 50% of states, though not the same 50% of states. All other internal components were *off* in all of the 64 states. In this case, as no lineage-specific TFs are expressed at any level, it is classified as a Th0 phenotype.

The computations to find the ergodic sets were implemented in PERL and were run on an 82-node Linux cluster. Most computations of the ergodic sets required around 10–20 gigabytes of RAM and took from hours to days for the Tarjan's algorithm to find an ergodic set. (Some required much more). In general, given an initial condition and *off*/*on* input composition, several ergodic sets could be reached. We found that out of the 512 possible input compositions, 502 compositions lead to a unique ergodic set and 6 of them lead to exactly two ergodic sets. There were three input compositions that led to one ergodic set, but for which the algorithm had not finished the complete search even after 7 days of calculations. Thus, for these three input compositions, there could be reachable ergodic sets that we did not identify. One input composition, in which all external inputs are active, ran for 7 days without finding any ergodic sets (this is the only input composition for which we have no ergodic set). As we got inconclusive results from the aforementioned incomplete analyses, the corresponding four input compositions were excluded from any reported results.

Model Simulations in Cell Collective

Model simulations were performed in the web-based modeling platform, Cell Collective[2]. Although the model is built by using discrete mathematics, the output activity levels of individual components can be represented as semi-continuous values ranging from 0 to 100% as previously described in Helikar et al. (2008) and Helikar and Rogers (2009). Each simulation was conducted using synchronous updates, and consisted of 5,000 steps, where the activity level of the measured output component was calculated as the fraction of ones (active states) over the last 500 iterations that describe the model's steady behavior (Helikar et al., 2008; Helikar and Rogers, 2009). The activity levels (dosage) of external components is unit-less and defined as a per-cent chance (probability * 100) of the component being active in a given time *t*. Depending on the desired experiment, the activity levels of external components can be set by the user to specific values, or they can be set to ranges from which values during each simulation are selected randomly (e.g., to simulate dose-response experiments).

Once the ergodic sets were identified, expressions of the internal components and their dependencies on the dosages of the external cytokines and the TCR ligand were investigated via the Cell Collective (Helikar et al., 2012b).

For each ergodic set, we chose one of its states as an initial condition and then simulated the model with the corresponding extracellular conditions via the Cell Collective. For each of the active input cytokines, the activity levels varied between 1 and 99%. Further details of the use of the Cell Collective are specific to the types of analysis as described below.

Sensitivity Analysis

The model was simulated in Cell Collective, whereby the activity levels of the inputs for each composition varied. By using the model-generated simulation data under 10,000 randomly generated environmental conditions, the association between inputs (cytokines and TCR ligand) and outputs (lineage-specifying TFs) was determined by probabilistic global sensitivity analysis based on PCC using the "*sensitivity*" package in R (R Development Core Team, 2011; Pujol et al., 2017). The PCC measures the strength of association between the output and input parameters after removing the linear effect of other input parameters (Marino et al., 2008; Pujol et al., 2017). The PCC between input and output is the correlation coefficients between residuals $(x_j - \hat{x}_j)$ and $(y - \hat{y})$, where x_j and y are input and output, respectively, and $\hat{x}_j\, \hat{y}$ are linear regression models [shown in Equation (1)] (Marino et al., 2008).

$$\hat{x}_j = c_0 + \sum_{\substack{p\ =\ 1 \\ p \neq j}}^{k} c_p x_p \text{ and } \hat{y} = b_0 + \sum_{\substack{p=1 \\ p \neq j}}^{k} b_p x_p. \qquad (1)$$

Optimal Settings Analysis

Once again, the model was simulated using 10,000 randomly generated environmental conditions for each input composition that can stimulate a complex phenotype. We sought to identify the environmental conditions wherein multiple lineage-specifying TFs can have balanced activity levels. First, we used the CV [Equation (2)] between TFs to measure variability. Further, we selected simulation results under which the lowest variability between TFs was observed. We selected corresponding environmental conditions that had lowest CV among TFs. Next, we selected the top 10 environmental conditions based on the outputs that have the highest activity levels of TFs. Thus, we considered both the balance of activity levels as wells as the quantity of co-expressed TFs. Finally, we defined ranges of activity levels of inputs from the selected environmental conditions. Further, for Th1–Th2, we simulated the effect of dominant inputs by individually varying IL-12, IL-18, IL-27, and the TCR ligand and using median activity levels from identified

[2]https://www.cellcollective.org

optimal activity levels for other inputs. We used R-scripts to determine the optimal activity levels from simulation data obtained via Cell Collective (Helikar et al., 2012b). The effect of dominant inputs on TFs in a complex phenotype was shown using the Generalized Additive Model (GAM) fitted scatter plots generated using "ggplot2" package in R.

$$\%CV = \frac{\text{Standard deviation}}{\text{mean}} \times 100. \quad (2)$$

AUTHOR CONTRIBUTIONS

BP, RT, DB, MB, and TH designed the research. BP, RT, AM, and DB performed the research. BP, MB, and TH analyzed the data. BP, RT, MB, and TH wrote the paper.

FUNDING

This project was supported by NIH grant no. 5R35GM119770-02 to TH and by the SILS Starting Grant of the University of Amsterdam to MB.

ACKNOWLEDGMENTS

We would like to thank Resa Helikar and Robert Moore for providing feedback on the manuscript.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphys.2018.00878/full#supplementary-material

REFERENCES

Abou-Jaoudé, W., Monteiro, P. T., Naldi, A., Grandclaudon, M., Soumelis, V., Chaouiya, C., et al. (2014). Model checking to assess T-helper cell plasticity. *Front. Bioeng. Biotechnol.* 2:86. doi: 10.3389/fbioe.2014.00086

Abou-Jaoudé, W., Traynard, P., Monteiro, P. T., Saez-Rodriguez, J., Helikar, T., Thieffry, D., et al. (2016). Logical modeling and dynamical analysis of cellular networks. *Front. Genet.* 7:94. doi: 10.3389/fgene.2016.00094

Aggarwal, S., Ghilardi, N., Xie, M. H., De Sauvage, F. J., and Gurney, A. L. (2003). Interleukin-23 promotes a distinct CD4 T cell activation state characterized by the production of interleukin-17. *J. Biol. Chem.* 278, 1910–1914. doi: 10.1074/jbc.M207577200

Aguado, E., Richelme, S., Nunez-Cruz, S., Miazek, A., Mura, A. M., Richelme, M., et al. (2002). Induction of T helper type 2 immunity by a point mutation in the LAT adaptor. *Science* 296, 2036–2040. doi: 10.1126/science.1069057

Albert, R., and Thakar, J. (2014). Boolean modeling: a logic-based dynamic approach for understanding signaling and regulatory networks and for making useful predictions. *Wiley Interdiscip. Rev. Syst. Biol. Med.* 6, 353–369. doi: 10.1002/wsbm.1273

Altin, J. A., Tian, L., Liston, A., Bertram, E. M., Goodnow, C. C., and Cook, M. C. (2011). Decreased T cell receptor signaling through CARD11 differentially compromises forkhead box protein 3-positive regulatory versus TH2 effector cells to cause allergy. *J. Allergy Clin. Immunol.* 127, 1277–1285. doi: 10.1016/j.jaci.2010.12.1081

Balázsi, G., Van Oudenaarden, A., and Collins, J. J. (2011). Cellular decision making and biological noise: from microbes to mammals. *Cell* 144, 910–925. doi: 10.1016/j.cell.2011.01.030

Barberis, M., Helikar, T., and Verbruggen, P. (2018). Simulation of stimulation: cytokine dosage and cell cycle crosstalk driving timing-dependent T cell differentiation. *Front. Physiol.* 9:879. doi: 10.3389/fphys.2018.00879

Barberis, M., Todd, R. G., and van der Zee, L. (2017). Advances and challenges in logical modeling of cell cycle regulation: perspective for multi-scale, integrative yeast cell models. *FEMS Yeast Res.* 17:fow103. doi: 10.1093/femsyr/fow103

Barberis, M., and Verbruggen, P. (2017). Quantitative systems biology to decipher design principles of a dynamic cell cycle network: the "Maximum Allowable mammalian Trade–Off–Weight" (MAmTOW). *NPJ Syst. Biol. Appl.* 3:26. doi: 10.1038/s41540-017-0028-x

Becskei, A., and Grusby, M. J. (2007). Contribution of IL-12R mediated feedback loop to Th1 cell differentiation. *FEBS Lett.* 581, 5199–5206. doi: 10.1016/j.febslet.2007.10.007

Bock, C. N., Babu, S., Breloer, M., Rajamanickam, A., Boothra, Y., Brunn, M.-L., et al. (2017). Th2/1 hybrid cells occurring in murine and human strongyloidiasis share effector functions of Th1 cells. *Front. Cell. Infect. Microbiol.* 7:261. doi: 10.3389/fcimb.2017.00261

Breitfeld, D., Ohl, L., Kremmer, E., Ellwart, J., Sallusto, F., Lipp, M., et al. (2000). Follicular B helper T cells express CXC chemokine receptor 5, localize to B cell follicles, and support immunoglobulin production. *J. Exp. Med.* 192, 1545–1552. doi: 10.1084/jem.192.11.1545

Carbo, A., Hontecillas, R., Kronsteiner, B., Viladomiu, M., Pedragosa, M., Lu, P., et al. (2013). Systems modeling of molecular mechanisms controlling cytokine-driven CD4 + T cell differentiation and phenotype plasticity. *PLoS Comput. Biol.* 9:e1003027. doi: 10.1371/journal.pcbi.1003027

Carbo, A., Olivares-Villagómez, D., Hontecillas, R., Bassaganya-Riera, J., Chaturvedi, R., Blanca Piazuelo, M., et al. (2014). Systems modeling of the role of interleukin-21 in the maintenance of effector CD4 + T cell responses during chronic *Helicobacter pylori* infection. *mBio* 5, 1–11. doi: 10.1128/mBio.01243-14

Chaouiya, C., Bérenguier, D., Keating, S. M., Naldi, A., van Iersel, M. P., Rodriguez, N., et al. (2013). SBML qualitative models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools. *BMC Syst. Biol.* 7:135. doi: 10.1186/1752-0509-7-135

Chen, L., and Flies, D. B. (2013). Molecular mechanisms of T cell co-stimulation and co-inhibition. *Nat. Rev. Immunol.* 13, 227–242. doi: 10.1038/nri3405

Chen, W., Jin, W., Hardegen, N., Lei, K., Li, L., Marinos, N., et al. (2003). Conversion of peripheral $CD4^+CD25^-$ naive T cells to $CD4^+CD25^+$ regulatory T cells by TGF-β induction of transcription factor *Foxp3*. *J. Exp. Med.* 198, 1875–1886. doi: 10.1084/jem.20030152

Chen, Z., Ding, J., Pang, N., Du, R., and Meng, W. (2013). The Th17/Treg balance and the expression of related cytokines in Uygur cervical cancer patients. *Diagn. Pathol.* 8:61. doi: 10.1186/1746-1596-8-61

Dardalhon, V., Awasthi, A., Kwon, H., Galileos, G., Gao, W., Sobel, R. A., et al. (2008). IL-4 inhibits TGF-beta-induced $Foxp3^+$ T cells and, together with TGF-beta, generates $IL\text{-}9^+IL\text{-}10^+$ Foxp3(-) effector T cells. *Nat. Immunol.* 9, 1347–1355. doi: 10.1038/ni.1677

Duan, M.-C., Han, W., Jin, P.-W., Wei, Y.-P., Wei, Q., Zhang, L.-M., et al. (2015). Disturbed Th17/Treg balance in patients with non-small cell lung cancer. *Inflammation* 38, 2156–2165. doi: 10.1007/s10753-015-0198-x

Eisenstein, E. M., and Williams, C. B. (2009). The Treg/Th17 cell balance: a new paradigm for autoimmunity. *Pediatr. Res.* 65, 26R–31R. doi: 10.1203/PDR.0b013e31819e76c7

Eizenberg-Magar, I., Rimer, J., Zaretsky, I., Lara-Astiaso, D., Reich-Zeliger, S., and Friedman, N. (2017). Diverse continuum of $CD4^+$ T cell states is determined by hierarchical additive integration of cytokine signals. *Proc. Natl. Acad. Sci. U.S.A.* 114, E6447–E6456. doi: 10.1073/pnas.1615590114

Evans, C. M., and Jenner, R. G. (2013). Transcription factor interplay in t helper cell differentiation. *Brief. Funct. Genomics* 12, 499–511. doi: 10.1093/bfgp/elt025

Fang, D., and Zhu, J. (2017). Dynamic balance between master transcription factors determines the fates and functions of CD4 T cell and innate lymphoid cell subsets. *J. Exp. Med.* 214, 1861–1876. doi: 10.1084/jem.20170494

Gosmann, C., Frazer, I. H., Mattarollo, S. R., and Blumenthal, A. (2014). IL-18, but not IL-12, induces production of IFN-γ in the immunosuppressive environment of HPV16 E7 transgenic hyperplastic skin. *J. Invest. Dermatol.* 134, 2562–2569. doi: 10.1038/jid.2014.201

Groux, H., O'Garra, A., Bigler, M., Rouleau, M., Antonenko, S., de Vries, J. E., et al. (1997). A $CD4^+$ T cell subset inhibits antigen-specific T cell responses and prevents colitis. *Nature* 389, 737–742. doi: 10.1038/39614

Harbour, S. N., Maynard, C. L., Zindl, C. L., Schoeb, T. R., and Weaver, C. T. (2015). Th17 cells give rise to Th1 cells that are required for the pathogenesis of colitis. *Proc. Natl. Acad. Sci. U.S.A.* 112, 7061–7066. doi: 10.1073/pnas.1415675112

Harrington, L. E., Hatton, R. D., Mangan, P. R., Turner, H., Murphy, T. L., Murphy, K. M., et al. (2005). Interleukin 17–producing $CD4^+$ effector T cells develop via a lineage distinct from the T helper type 1 and 2 lineages. *Nat. Immunol.* 6, 1123–1132. doi: 10.1038/ni1254

Harrison, O. J., Srinivasan, N., Pott, J., Schiering, C., Krausgruber, T., Ilott, N. E., et al. (2015). Epithelial-derived IL-18 regulates Th17 cell differentiation and $Foxp3^+$ Treg cell function in the intestine. *Mucosal Immunol.* 8, 1226–1236. doi: 10.1038/mi.2015.13

Hegazy, A. N., Peine, M., Helmstetter, C., Panse, I., Fröhlich, A., Bergthaler, A., et al. (2010). Interferons direct Th2 cell reprogramming to generate a stable GATA-3^+ T-bet + Cell Subset with Combined Th2 and Th1 Cell Functions. *Immunity* 32, 116–128. doi: 10.1016/j.immuni.2009.12.004

Helikar, T., Konvalina, J., Heidel, J., and Rogers, J. A. (2008). Emergent decision-making in biological signal transduction networks. *Proc. Natl. Acad. Sci. U.S.A.* 105:705088105. doi: 10.1073/pnas.0705088105

Helikar, T., Kowal, B., Madrahimov, A., Shrestha, M., Pedersen, J., Limbu, K., et al. (2012a). Bio-logic builder: a Non-Technical tool for building dynamical, qualitative models. *PLoS One* 7:e46417. doi: 10.1371/journal.pone.0046417

Helikar, T., Kowal, B., McClenathan, S., Bruckner, M., Rowley, T., Madrahimov, A., et al. (2012b). The cell collective: toward an open and collaborative approach to systems biology. *BMC Syst. Biol.* 6:96. doi: 10.1186/1752-0509-6-96

Helikar, T., Kowal, B., and Rogers, J. A. (2013). A cell simulator platform: the cell collective. *Clin. Pharmacol. Ther.* 93, 393–395. doi: 10.1038/clpt.2013.41

Helikar, T., and Rogers, J. A. (2009). ChemChains: a platform for simulation and analysis of biochemical networks aimed to laboratory scientists. *BMC Syst. Biol.* 3:58. doi: 10.1186/1752-0509-3-58

Ichiyama, K., Yoshida, H., Wakabayashi, Y., Chinen, T., Saeki, K., Nakaya, M., et al. (2008). Foxp3 inhibits RORgammat-mediated IL-17A mRNA transcription through direct interaction with RORgammat. *J. Biol. Chem.* 283, 17003–17008. doi: 10.1074/jbc.M801286200

Kaiko, G. E., Horvat, J. C., Beagley, K. W., and Hansbro, P. M. (2008). Immunological decision-making: how does the immune system decide to mount a helper T cell response? *Immunology* 123, 326–338. doi: 10.1111/j.1365-2567.2007.02719.x

Kaplan, M. H. (2013). Th9 cells: differentiation and disease. *Immunol. Rev.* 252, 104–115. doi: 10.1111/imr.12028

Kimura, A., and Kishimoto, T. (2010). IL-6: regulator of Treg/Th17 balance. *Eur. J. Immunol.* 40, 1830–1835. doi: 10.1002/eji.201040391

Koch, M. A., Tucker-Heard, G., Perdue, N. R., Killebrew, J. R., Urdahl, K. B., and Campbell, D. J. (2009). The transcription factor T-bet controls regulatory T cell homeostasis and function during type 1 inflammation. *Nat. Immunol.* 10, 595–602. doi: 10.1038/ni.1731

Kullberg, M. C., Jankovic, D., Feng, C. G., Hue, S., Gorelick, P. L., McKenzie, B. S., et al. (2006). IL-23 plays a key role in *Helicobacter hepaticus*-induced T cell-dependent colitis. *J. Exp. Med.* 203, 2485–2494. doi: 10.1084/jem.20061082

Le Novère, N. (2015). Quantitative and logic modelling of molecular and gene networks. *Nat. Rev. Genet.* 16, 146–158. doi: 10.1038/nrg3885

Lee, Y. K., Mukasa, R., Hatton, R. D., and Weaver, C. T. (2009a). Developmental plasticity of Th17 and Treg cells. *Curr. Opin. Immunol.* 21, 274–280. doi: 10.1016/j.coi.2009.05.021

Lee, Y. K., Turner, H., Maynard, C. L., Oliver, J. R., Chen, D., Elson, C. O., et al. (2009b). Late Developmental Plasticity in the T Helper 17 Lineage. *Immunity* 30, 92–107. doi: 10.1016/j.immuni.2008.11.005

Linke, C., Chasapi, A., González-Novo, A., Al Sawad, I., Tognetti, S., Klipp, E., et al. (2017). A Clb/Cdk1-mediated regulation of Fkh2 synchronizes CLB expression in the budding yeast cell cycle. *NPJ Syst. Biol. Appl.* 3:7. doi: 10.1038/s41540-017-0008-1

Luchting, B., Rachinger-Adam, B., Zeitler, J., Egenberger, L., Möhnle, P., Kreth, S., et al. (2014). Disrupted TH17/Treg balance in patients with chronic low back pain. *PLoS ONE* 9:e104883. doi: 10.1371/journal.pone.0104883

Ma, C. S., Deenick, E. K., Batten, M., and Tangye, S. G. (2012). The origins, function, and regulation of T follicular helper cells. *J. Exp. Med.* 209, 1241–1253. doi: 10.1084/jem.20120994

Mangan, P. R., Harrington, L. E., O'Quinn, D. B., Helms, W. S., Bullard, D. C., Elson, C. O., et al. (2006). Transforming growth factor-β induces development of the TH17 lineage. *Nature* 441, 231–234. doi: 10.1038/nature04754

Marino, S., Hogue, I. B., Ray, C. J., and Kirschner, D. E. (2008). A methodology for performing global uncertainty and sensitivity analysis in systems biology. *J. Theor. Biol.* 254, 178–196. doi: 10.1016/j.jtbi.2008.04.011

Martinez-Sanchez, M. E., Huerta, L., Alvarez-Buylla, E. R., and Villarreal Lujan, C. (2018). Role of cytokine combinations on CD4+ T cell differentiation, partial polarization, and plasticity: continuous network modeling approach. *Front. Physiol.* 9:877. doi: 10.3389/fphys.2018.00877

Martinez-Sanchez, M. E., Mendoza, L., Villarreal, C., and Alvarez-Buylla, E. R. (2015). A Minimal Regulatory Network of Extrinsic and Intrinsic Factors Recovers Observed Patterns of CD4 + T Cell Differentiation and Plasticity. *PLoS Comput. Biol.* 11:e1004324. doi: 10.1371/journal.pcbi.1004324

Morrison, P. J., Bending, D., Fouser, L. A., Wright, J. F., Stockinger, B., Cooke, A., et al. (2013). Th17-cell plasticity in *Helicobacter hepaticus*-induced intestinal inflammation. *Mucosal Immunol.* 6, 1143–1156. doi: 10.1038/mi.2013.11

Mosmann, T. R., Cherwinski, H., Bond, M. W., Giedlin, M. A., and Coffman, R. L. (1986). Two types of murine helper T cell clone. I. Definition according to profiles of lymphokine activities and secreted proteins. *J. Immunol.* 136, 2348–2357. doi: 10.1111/j.1442-9071.2011.02672.x

Munk, R. B., Sugiyama, K., Ghosh, P., Sasaki, C. Y., Rezanka, L., Banerjee, K., et al. (2011). Antigen-independent IFN-γ production by human naïve $CD4^+$ T cells activated by IL-12 plus IL-18. *PLoS One* 6:e18553. doi: 10.1371/journal.pone.0018553

Murai, H., Qi, H., Choudhury, B., Wild, J., Dharajiya, N., Vaidya, S., et al. (2012). Alternaria-induced release of IL-18 from damaged airway epithelial cells: an NF-??B dependent mechanism of Th2 differentiation? *PLoS One* 7:e30280. doi: 10.1371/journal.pone.0030280

Murphy, K. M., and Reiner, S. L. (2002). The lineage decisions of helper T cells. *Nat. Rev. Immunol.* 2, 933–944. doi: 10.1038/nri954

Nakanishi, K., Yoshimoto, T., Tsutsui, H., and Okamura, H. (2001). Interleukin-18 regulates both Th1 and Th2 responses. *Annu. Rev. Immunol.* 19, 423–474. doi: 10.1146/annurev.immunol.19.1.423

Naldi, A., Carneiro, J., Chaouiya, C., and Thieffry, D. (2010). Diversity and plasticity of Th cell types predicted from regulatory network modelling. *PLoS Comput. Biol.* 6:e1000912. doi: 10.1371/journal.pcbi.1000912

Naldi, A., Monteiro, P. T., Müssel, C., Kestler, H. A., Thieffry, D., Xenarios, I., et al. (2015). Cooperative development of logical modelling standards and tools with CoLoMoTo. *Bioinformatics* 31, 1154–1159. doi: 10.1093/bioinformatics/btv013

Nindl, V., Maier, R., Ratering, D., De Giuli, R., Züst, R., Thiel, V., et al. (2012). Cooperation of Th1 and Th17 cells determines transition from autoimmune myocarditis to dilated cardiomyopathy. *Eur. J. Immunol.* 42, 2311–2321. doi: 10.1002/eji.201142209

Ohnmacht, C., Park, J., Cording, S., Wing, J. B., Atarashi, K., Obata, Y., et al. (2015). The microbiota regulates type 2 immunity through RORγt + T cells. *Science* 349, 1–9. doi: 10.1126/science.aac4263

Oldenhove, G., Bouladoux, N., Wohlfert, E. A., Hall, J. A., Chou, D., Dos Santos, L., et al. (2009). Decrease of Foxp3 + Treg cell number and acquisition of effector cell phenotype during lethal infection. *Immunity* 31, 772–786. doi: 10.1016/j.immuni.2009.10.001

Omenetti, S., and Pizarro, T. T. (2015). The Treg/Th17 axis: a dynamic balance regulated by the gut microbiome. *Front. Immunol.* 6:639. doi: 10.3389/fimmu.2015.00639

Palau-Ortin, D., Formosa-Jordan, P., Sancho, J. M., and Ibañes, M. (2015). Pattern selection by dynamical biochemical signals. *Biophys. J.* 108, 1555–1565. doi: 10.1016/j.bpj.2014.12.058

Park, H., Li, Z., Yang, X. O., Chang, S. H., Nurieva, R., Wang, Y.-H., et al. (2005). A distinct lineage of CD4 T cells regulates tissue inflammation by producing interleukin 17. *Nat. Immunol.* 6, 1133–1141. doi: 10.1038/ni1261

Paul, W. E. (2010). What determines Th2 differentiation, in vitro and in vivo? *Immunol. Cell Biol.* 88, 236–239. doi: 10.1038/icb.2010.2

Peine, M., Rausch, S., Helmstetter, C., Fröhlich, A., Hegazy, A. N., Kühl, A. A., et al. (2013). Stable T-bet + GATA-3 + Th1/Th2 hybrid cells arise in vivo, can develop directly from naive precursors, and limit immunopathologic inflammation. *PLoS Biol.* 11:e1001633. doi: 10.1371/journal.pbio.1001633

Podojil, J. R., and Miller, S. D. (2009). Molecular mechanisms of T cell receptor and costimulatory molecule ligation/blockade in autoimmune disease therapy. *Immunol. Rev.* 229, 337–355. doi: 10.1111/j.1600-065X.2009.00773.x

Pujol, G., Iooss, B., Alexandre Janon with contributions from Khalid Boumhaout, Da Veiga, S. A., Delage, T., Fruth, J., et al. (2017). *Sensitivity: Global Sensitivity Analysis of Model Outputs*. Available at: https://cran.r-project.org/package=sensitivity

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. Vienna: R Foundation for Statistical Computing, doi: 10.1007/978-3-540-74686-7

Reiner, S. L. (2007). Development in motion: helper T cells at work. *Cell* 129, 33–36. doi: 10.1016/j.cell.2007.03.019

Ribeiro, A. S., and Kauffman, S. A. (2007). Noisy attractors and ergodic sets in models of gene regulatory networks. *J. Theor. Biol.* 247, 743–755. doi: 10.1016/j.jtbi.2007.04.020

Romagnani, S. (2000). T cell subsets (Th1 versus Th2). *Ann. Allergy Asthma Immunol.* 85, 9–18; quiz 18, 21. doi: 10.1016/S1081-1206(10)62426-X

Rowell, E., and Wilson, C. B. (2009). Programming perpetual T helper cell plasticity. *Immunity* 30, 7–9. doi: 10.1016/j.immuni.2008.12.012

Samaga, R., and Klamt, S. (2013). Modeling approaches for qualitative and semi-quantitative analysis of cellular signaling networks. *Cell Commun. Signal.* 11:43. doi: 10.1186/1478-811X-11-43

Schaerli, P., Willimann, K., Lang, A. B., Lipp, M., Loetscher, P., and Moser, B. (2000). CXC chemokine receptor 5 expression defines follicular homing T cells with B cell helper function. *J. Exp. Med.* 192, 1553–1562. doi: 10.1084/jem.192.11.1553

Schmitt, E. G., and Williams, C. B. (2013). Generation and function of induced regulatory T cells. *Front. Immunol.* 4:152. doi: 10.3389/fimmu.2013.00152

Shi, G., Cox, C. A., Vistica, B. P., Tan, C., Wawrousek, E. F., and Gery, I. (2008). Phenotype switching by inflammation-inducing polarized Th17 cells, but not by Th1 cells. *J. Immunol.* 181, 7205–7213. doi: 10.4049/jimmunol.181.10.7205

Soroosh, P., and Doherty, T. A. (2009). Th9 and allergic disease. *Immunology* 127, 450–458. doi: 10.1111/j.1365-2567.2009.03114.x

Tarjan, R. (1972). Depth-first search and linear graph algorithms. *SIAM J. Comput.* 1, 146–160. doi: 10.1137/0201010

Tartar, D. M., VanMorlan, A. M., Wan, X., Guloglu, F. B., Jain, R., Haymaker, C. L., et al. (2010). FoxP3$^+$ ROR t + T helper intermediates display suppressive function against autoimmune diabetes. *J. Immunol.* 184, 3377–3385. doi: 10.4049/jimmunol.0903324

Tesmer, L. A., Lundy, S. K., Sarkar, S., and Fox, D. A. (2008). Th17 cells in human disease. *Immunol. Rev.* 223, 87–113. doi: 10.1111/j.1600-065X.2008.00628.x

Tijms, H. C. (2003). *A First Course in Stochastic Models*. New York, NY: John Wiley & Sons, Ltd, doi: 10.1002/047001363X

Todd, R. G., and Helikar, T. (2012). Ergodic sets as cell phenotype of budding yeast cell cycle. *PLoS One* 7:e45780. doi: 10.1371/journal.pone.0045780

Tominaga, K., Yoshimoto, T., Torigoe, K., Kurimoto, M., Matsui, K., Hada, T., et al. (2000). IL-12 synergizes with IL-18 or IL-1beta for IFN-gamma production from human T cells. *Int. Immunol.* 12, 151–160. doi: 10.1093/intimm/12.2.151

van Panhuys, N., Klauschen, F., and Germain, R. N. (2014). T cell-receptor-dependent signal intensity dominantly controls CD4 + T cell polarization in vivo. *Immunity* 41, 63–74. doi: 10.1016/j.immuni.2014.06.003

Veldhoen, M., Uyttenhove, C., van Snick, J., Helmby, H., Westendorf, A., Buer, J., et al. (2008). Transforming growth factor-beta "reprograms" the differentiation of T helper 2 cells and promotes an interleukin 9-producing subset. *Nat. Immunol.* 9, 1341–1346. doi: 10.1038/ni.1659

Voo, K. S., Wang, Y.-H., Santori, F. R., Boggiano, C., Wang, Y.-H., Arima, K., et al. (2009). Identification of IL-17-producing FOXP3 + regulatory T cells in humans. *Proc. Natl. Acad. Sci. U.S.A.* 106, 4793–4798. doi: 10.1073/pnas.0900408106

Xu, L., Kitani, A., Fuss, I., and Strober, W. (2007). Cutting edge: regulatory T cells induce CD4$^+$ CD25-Foxp3- T cells or are self-induced to become Th17 cells in the absence of exogenous TGF-beta. *J. Immunol.* 178, 6725–6729. doi: 10.4049/jimmunol.178.11.6725

Yamane, H., and Paul, W. E. (2012). Cytokines of the γc family control CD4 + T cell differentiation and function. *Nat. Immunol.* 13, 1037–1044. doi: 10.1038/ni.2431

Yamane, H., Zhu, J., and Paul, W. E. (2005). Independent roles for IL-2 and GATA-3 in stimulating naive CD4 + T cells to generate a Th2-inducing cytokine environment. *J. Exp. Med.* 202, 793–804. doi: 10.1084/jem.20051304

Yang, X. O., Nurieva, R., Martinez, G. J., Kang, H. S., Chung, Y., Pappu, B. P., et al. (2008). Molecular antagonism and plasticity of regulatory and inflammatory T cell programs. *Immunity* 29, 44–56. doi: 10.1016/j.immuni.2008.05.007

Yoshimoto, T., Takeda, K., Tanaka, T., Ohkusu, K., Kashiwamura, S., Okamura, H., et al. (1998). IL-12 up-regulates IL-18 receptor expression on T cells, Th1 cells, and B cells: synergism with IL-18 for IFN-gamma production. *J. Immunol.* 161, 3400–3407.

Yu, F., Sharma, S., Edwards, J., Feigenbaum, L., and Zhu, J. (2015). Dynamic expression of transcription factors T-bet and GATA-3 by regulatory T cells maintains immunotolerance. *Nat. Immunol.* 16, 197–206. doi: 10.1038/ni.3053

Zhou, L., Lopes, J. E., Chong, M. M. W., Ivanov, I. I., Min, R., Victora, G. D., et al. (2008). TGF-β-induced Foxp3 inhibits TH17 cell differentiation by antagonizing RORγt function. *Nature* 453, 236–240. doi: 10.1038/nature06878

Zhu, J., and Paul, W. E. (2008). CD4 T cells: fates, functions, and faults. *Blood* 112, 1557–1569. doi: 10.1182/blood-2008-05-078154

Zhu, J., Yamane, H., and Paul, W. E. (2010). Differentiation of effector CD4 T cell populations. *Annu. Rev. Immunol.* 28, 445–489. doi: 10.1146/annurev-immunol-030409-101212

Conflict of Interest Statement: TH has served as a scientific advisor and/or consultant to Discovery Collective.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

ORIGINAL RESEARCH
published: 02 August 2018
doi: 10.3389/fphys.2018.00877

Role of Cytokine Combinations on CD4+ T Cell Differentiation, Partial Polarization, and Plasticity: Continuous Network Modeling Approach

Mariana E. Martinez-Sanchez[1,2], Leonor Huerta[3], Elena R. Alvarez-Buylla[1,2] and Carlos Villarreal Luján[2,4]**

[1] Laboratorio Genética Molecular, Epigenética, Desarrollo y Evolución de Plantas, Departamento de Ecología Funcional, Instituto de Ecología, Universidad Nacional Autónoma de México, Mexico City, Mexico, [2] Centro de Ciencias de la Complejidad, Universidad Nacional Autónoma de México, Mexico City, Mexico, [3] Laboratorio B108, Departmento de Immunología, Instituto de Investigaciones Biomédicas, Universidad Nacional Autónoma de México, Mexico City, Mexico, [4] Departamento de Física Cuántica y Fotónica, Instituto de Física, Universidad Nacional Autónoma de México, Mexico City, Mexico

OPEN ACCESS

Edited by:
Doron Levy,
University of Maryland, College Park, United States

Reviewed by:
Syed Aun Muhammad,
Bahauddin Zakariya University, Pakistan
Paolo Tieri,
Consiglio Nazionale Delle Ricerche (CNR), Italy

***Correspondence:**
Elena R. Alvarez-Buylla
eabuylla@gmail.com
Carlos Villarreal Luján
carlos@fisica.unam.mx

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: 11 December 2017
Accepted: 19 June 2018
Published: 02 August 2018

Citation:
Martinez-Sanchez ME, Huerta L, Alvarez-Buylla ER and Villarreal Luján C (2018) Role of Cytokine Combinations on CD4+ T Cell Differentiation, Partial Polarization, and Plasticity: Continuous Network Modeling Approach. Front. Physiol. 9:877. doi: 10.3389/fphys.2018.00877

Purpose: We put forward a theoretical and dynamical approach for the semi-quantitative analysis of CD4+ T cell differentiation, the process by which cells with different functions are derived from activated $CD4^+$ T naïve lymphocytes in the presence of particular cytokine microenvironments. We explore the system-level mechanisms that underlie CD4+ T plasticity—the conversion of polarized cells to phenotypes different from those originally induced.

Methods: In this paper, we extend a previous study based on a Boolean network to a continuous framework. The network includes transcription factors, signaling pathways, as well as autocrine and exogenous cytokines, with interaction rules derived using fuzzy logic.

Results: This approach allows us to assess the effect of relative differences in the concentrations and combinations of exogenous and endogenous cytokines, as well as of the expression levels of diverse transcription factors. We found either abrupt or gradual differentiation patterns between observed phenotypes depending on critical concentrations of single or multiple environmental cytokines. Plastic changes induced by environmental cytokines were observed in conditions of partial phenotype polarization in the T helper 1 to T helper 2 transition. On the other hand, the T helper 17 to induced regulatory T-cells transition was highly dependent on cytokine concentrations, with TGFβ playing a prime role.

Conclusion: The present approach is useful to further understand the system-level mechanisms underlying observed patterns of $CD4^+$ T differentiation and response to changing immunological challenges.

Keywords: CD4+ T cells, regulatory network, ODE, heterogeneity, plasticity, micro-environment, cytokines

INTRODUCTION

The phenotype of a cell emerges from the feedback between internal regulatory networks and signals from the microenvironment (Murphy and Stockinger, 2010; DuPage and Bluestone, 2016). CD4+ T cells constitute a useful model to evaluate the role of micro-environmental signals on intracellular regulatory networks underlying cell differentiation and plasticity, as the combination and concentration of exogenous cytokines are crucial for CD4+ T cell differentiation and plasticity (Murphy and Stockinger, 2010; DuPage and Bluestone, 2016; Eizenberg-Magar et al., 2017).

CD4+ T cells are part of the adaptive immune response. Naïve CD4+ T cells are activated in response to antigens presented by antigen presenting cells (APC) (Zhu et al., 2010). Depending on the cytokines in the microenvironment, these cells may differentiate into particular subsets. APCs are the main source of cytokines (extrinsic cytokines) initiating an immune response, but they can also be produced by other cells of the organism (Duque and Descoteaux, 2014; Sozzani et al., 2017). Exogenous cytokines bind to the membrane receptors of the cell and activate intracellular signaling pathways. These signals activate or inhibit particular transcription factors integrated in the networks under analysis and promote the production of autocrine cytokines, creating a positive feedback that reinforces the polarization dynamics (Zhu et al., 2010). In addition, autocrine cytokines that can also activate or inhibit other cells of the immune system. It is interesting to note that different cytokines combinations have been shown to have synergistic or antagonistic effects on CD4+ T cell differentiation, and such differential responses may be crucial during immune responses to pathogen attack, modulation of the immune response, or immunopathological conditions (Zhu et al., 2010).

Functional CD4+ T lymphocytes can be grouped into subsets known as Th1, Th2, Th3, Th9, Th17, Treg, Tr1, and Tfh (**Table 1**). It has been documented that Th1 cells require extrinsic IL-12 and IFNγ, they express T-bet and IFNγ (Hsieh et al., 1993; Perez et al., 1995; Szabo et al., 2000, 2003). Th2 cells require extrinsic IL-4 and are stabilized by IL-2, they express GATA3, IL-4, IL-5, and -IL13 (Le Gros et al., 1990; Swain et al., 1990; Cote-Sierra et al., 2004; Ansel et al., 2006; Zheng and Flavell, 1997). Th3 cells require extrinsic TGFβ and express TGFβ (Gol-Ara et al., 2012). Th9 cells require IL-4 and TGFβ, they express IL-9 (Lu et al., 2012; Kaplan, 2013; Schmitt et al., 2014). Th17 cells require extrinsic TGFβ and IL-6, IL-21 or IL-23, they produce RORγt, IL-21, IL-17A, and IL-17F (Ivanov et al., 2006; Veldhoen et al., 2006; Zhou et al., 2007; Korn et al., 2009). Treg cells require extrinsic TGFβ and IL-2, they express Foxp3, TGFβ and in some cases IL-10 (Chen et al., 2003; Hori et al., 2003; Davidson et al., 2007; Zheng et al., 2007). Tr1 cells require extrinsic IL10, expressing IL10 (Roncarolo et al., 2006; Awasthi et al., 2007; Gagliani et al., 2015).Tfh cells require IL-21, they express Bcl6 (Johnston et al., 2009; Nurieva et al., 2009; Yu et al., 2009; Crotty, 2014).

TABLE 1 | CD4+ T cell types, their associated transcription factors, characteristic cytokines, and exogenous cytokines that induce the cell type.

Cell type	Transcription factor	Characteristic cytokines	Induced by
Th1	T-bet	IFNγ	IFNγ, IL-12
Th2	GATA3	IL-4	IL-4, IL-2
Th17	RORγt	IL-17, IL-21	TGFβ, IL-6, IL-21
Tfh	Bcl6	IL-21	IL-21
Th9	–	IL-9	TGFβ, IL-4
iTreg	Foxp3	TGFβ	TGFβ, IL-2
Tr1	–	IL-10	IL-10, IL-27
Th3	–	TGFβ	TGFβ

Furthermore, CD4+ T cells are highly heterogeneous suggesting that cell populations go through a continuum of polarization levels after initial priming (Murphy and Stockinger, 2010; Magombedze et al., 2013; DuPage and Bluestone, 2016; Eizenberg-Magar et al., 2017). Thus, mixed cellular phenotypes may be encountered under particular cytokine concentrations and combinations, and in some cases, hybrid cell types such as Th1-like and Th2-like regulatory cells or Th1/Th2 hybrids have been documented (Koch et al., 2009; Hegazy et al., 2010; Wohlfert et al., 2011). Studies performed on polarized CD4+ T cell populations indicate that, even under controlled *in vitro* conditions, stimulation generates heterogeneous cell populations with variable cytokine expression profiles or intermediate cell types (Assenmacher et al., 1994; Bucy et al., 1994; Openshaw et al., 1995; Kelso et al., 1999; Chang et al., 2007; Eizenberg-Magar et al., 2017). Asymmetric cell division with segregation of signaling proteins may explain this behavior (Verbist et al., 2016).

The same cytokines responsible for the induction of naïve cells to a particular polarized state may also dictate the conversion from a different subset to this state. For example, multiple studies report the transit of Treg cells toward Th17 cells in response to the addition of exogenous IL-6 in the presence of TGFβ (Yang et al., 2008; Lee et al., 2009a; Murphy and Stockinger, 2010). Other plastic transitions depend on the degree of polarization, as in the case of the Th17/Treg (Michalek et al., 2011; Berod et al., 2014; Gagliani et al., 2015) and the Th1/Th2 transition (Perez et al., 1995; Murphy et al., 1996; Hegazy et al., 2010). Recently polarized Th1 and Th2 cells can transdifferentiate into other subsets in response to environmental IL-4 or IL-12, but fully polarized Th1 and Th2 cells are robust and do not change their state in response to different microenvironments (Murphy et al., 1996). Despite abundant experimental data on such rich differentiation and plastic responses of CD4+ T cells in contrasting microenvironments, we still do not understand the underlying system-level mechanisms that explain such responses. To contribute in this direction our group and others have been integrating complex multistable regulatory network models that have been partially validated with experimental data (Mendoza, 2006; Naldi et al., 2010; Carbo et al., 2013; Abou-Jaoudé et al., 2014; Martinez-Sanchez et al., 2015; Eizenberg-Magar et al., 2017).

Complex regulatory networks are useful to model multistability, as they reach different stable multidimensional configurations, called attractors that correspond to expression profiles of different cell types (Kauffman, 1969; Mendoza et al., 1999; Bornholdt, 2008; Villarreal et al., 2012;

Martínez-Sosa and Mendoza, 2013; Albert and Thakar, 2014; Naldi et al., 2015; Alvarez-Buylla et al., 2016). Hence, this type of models have been used in other systems to successfully explore the system-level mechanisms underlying cell differentiation (Kauffman, 1969; Mendoza et al., 1999; Bornholdt, 2008; Cortes et al., 2008; Azpeitia et al., 2011, 2014; Villarreal et al., 2012; Martínez-Sosa and Mendoza, 2013; Albert and Thakar, 2014; Naldi et al., 2015; Alvarez-Buylla et al., 2016; Davila-Velderraín et al., 2017). We previously proposed a Boolean network model that incorporates critical components to study CD4+ T cell subsets differentiation and plasticity (Martinez-Sanchez et al., 2015). In the present paper we have extended the Boolean model to a system with network interactions defined by fuzzy logic propositions. In this kind of approach, a fuzzy variable may acquire truth values within the continuous range [0,1]. The dynamic evolution of the network relations are described by a set of ordinary of differential equations (ODE) that enables us to analyze the role of alterations on cytokines concentrations and combinations, as well as other system's components modifications on CD4+ T cell differentiation and plasticity. Each cell state or type corresponds to an attractor, and our system let us to study the conditions required to drive the system from one attractor to another one (Haken, 1977). We explore pathways that lead to equilibrium points, but also alterations of the expression levels of the networks components and the microenvironment, that may induce that cells transit between attractors (Mendoza, 2006; Naldi et al., 2010; Carbo et al., 2013; Abou-Jaoudé et al., 2014; Martinez-Sanchez et al., 2015; Eizenberg-Magar et al., 2017; Barberis et al., 2018; Puniya et al., 2018).

The continuous network model proposed here allows semi-quantitative evaluations of alterations of the inputs (exogenous cytokines) and the intrinsic components (transcription factors, signaling pathways, and autocrine cytokines) on cell-type transitions (Villarreal et al., 2012; Davila-Velderrain et al., 2015). The study involves an adaptation of a method specifically designed to study the so-called epigenetic landscape repatterning under altered microenvironmental conditions (Davila-Velderrain et al., 2015; Perez-Ruiz et al., 2015). Our model involves a set of regulatory interactions results that reproduce the main polarized phenotypes of CD4+ T cells and several of the plasticity patterns reported in the experimental literature. We determine the effect of systematic changes in the concentrations of exogenous cytokines and the internal state of the network in the differentiation and plasticity of CD4+ T cells. We focus on the Th1/Th2, and Th17/iTreg transitions, given that these have been thoroughly characterized, due to their pathogenic and therapeutic relevance (DuPage and Bluestone, 2016). This approach uncovers the signaling circuitry underlying the robust fully polarized Th1 and Th2 responses, and predicts that the phenotypic shift from a cell-mediated cytotoxic to a humoral immune response is possible only in early stages of CD4+ T cell differentiation. It also shows that a shift from inflammatory to induced regulatory immune response is much less restrictive. This finding and the overall framework put forward here may be useful to further understand the systemic mechanisms underlying immunological diseases where cellular plasticity plays a prime role (DuPage and Bluestone, 2016).

MATERIALS AND METHODS

Network Construction

We constructed the CD4+ T cell regulatory network using available experimental data (**Figure 1A**). The network includes nodes that correspond to transcription factors, signal transduction pathway components, and cytokine receptors, as well as autocrine and exogenous cytokines. The edges of the network correspond to the verified regulatory interactions between the nodes (**Supplementary Data Sheets S1**, **S2**) (Martinez-Sanchez et al., 2015). The value of the node depends on the state of its regulators defined by a logical rule (**Figure 1B**). In the Boolean approach, each node of the network has a value that corresponds to its expression level, where 0 corresponds to the basal level of expression (inactive) and 1 to the maximum normalized expression level (active), while in the continuous model the value of each node is a real number in the range [0,1]. The model was validated by verifying that the predicted CD4+ T cell subsets and plasticity transitions coincide with experimental observations (**Figure 2** and **Supplementary Data Sheet S2**) (Martinez-Sanchez et al., 2015).

The final network consists of 21 nodes (**Figure 2**). Five nodes correspond to transcription factors (TBET, GATA3, FOXP3, RORGT, and BCL6); seven nodes correspond to signaling pathways integrating signal transducers such as STAT proteins, interleukin receptors, and autocrine cytokines (IFNG, IL2, IL4, IL10, TGFB, IL9, and IL21); nine nodes correspond to exogenous cytokines, that are produced by other cells of the immune system and thus act as inputs to the network (IFNGe, IL12e, IL2e, IL4e, IL10e, IL27e, TGFBe, IL6e, and IL21e). These are marked with an "e" (exogenous) after the cytokine name. To study the effect of the microenvironment we focused on nine biologically relevant environments (Zhu et al., 2010): pro-Th0, pro-Th1, pro-Th2, pro-Th17, pro-Th9, pro-Tfh, pro-iTreg, pro-Tr1, and pro-Th3 (**Table 1**). The regulatory cytokine IL-10 deserves special consideration, since it uses STAT3, similarly as IL-2 and the inflammatory cytokines IL-6. Thus, we assume that IL-10 signaling is mediated by an independent pathway, different from that of IL-6/IL-21, even though they share STAT3 as a messenger molecule (Moore et al., 2001). While IL-27 has been linked to multiple functions, we consider that its main role in the model is regulatory (Awasthi et al., 2007; Murugaiyan et al., 2009; Pot et al., 2009). The model ignores weak interactions, chemokines, and epigenetic regulation that are also relevant and should be included in future modeling efforts.

Fuzzy Logic Approach

The Boolean scheme allows to establish the main topological features of the network interactions; however, it only includes variables with dichotomous values. A more realistic approach should consider that variables and parameters with a continuous range of expression values. With that purpose we propose a model based on fuzzy logic where, not only the variables, but

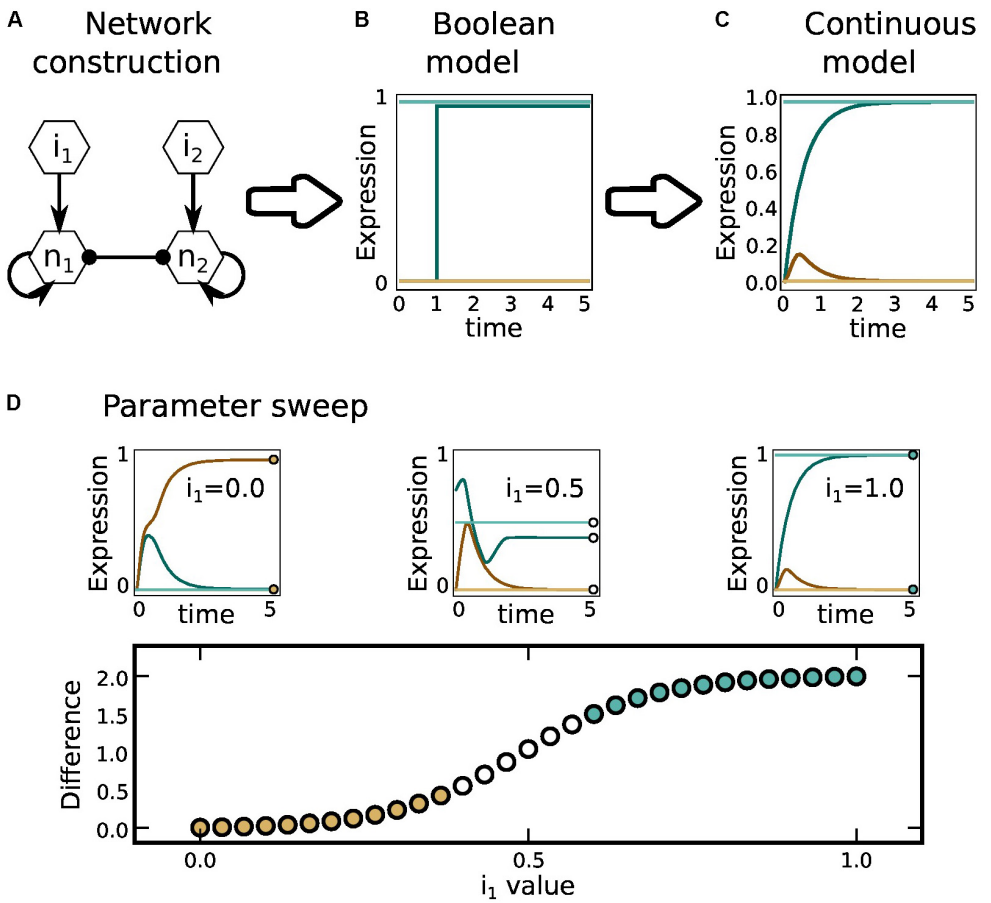


FIGURE 1 | Methodology. Using available experimental data we constructed **(A)** the regulatory network, **(B)** Boolean functions (Martinez-Sanchez et al., 2015), and **(C)** ordinary differential equations (current article). **(D)** We then determined the resulting steady state for different concentrations and exogenous cytokines.

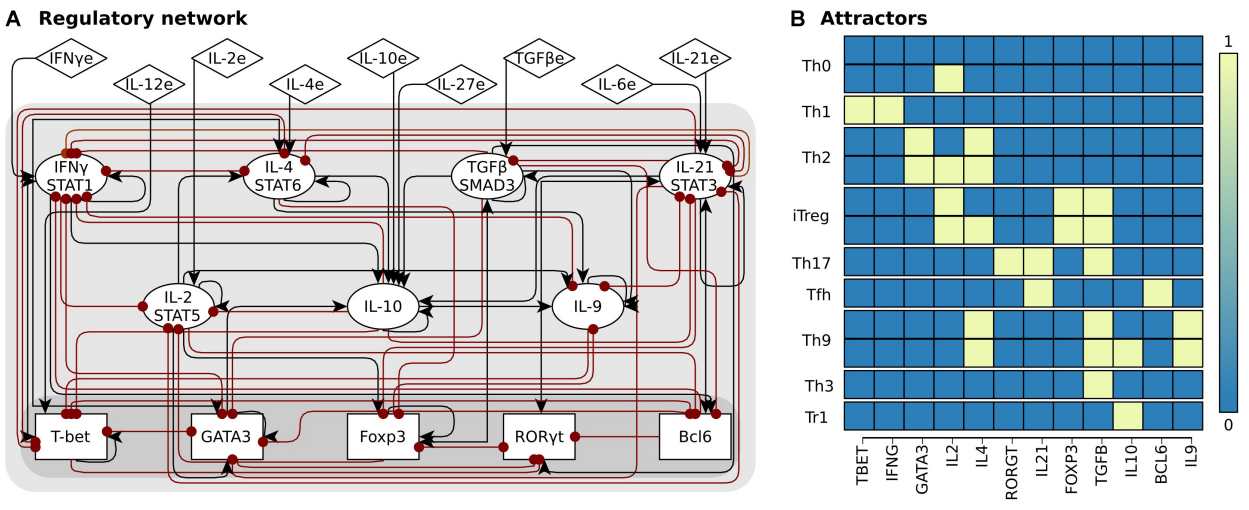


FIGURE 2 | CD4+ T cell transcriptional-signaling regulatory network. The regulatory network was constructed using available experimental data. **(A)** The network includes transcription factors (rectangles), autocrine cytokines and their signaling pathways (ellipses) and exogenous cytokines (diamonds). Interactions leading to activation are represented by black arrows, while those leading to inhibition with red dots. **(B)** Sample attractors of the system.

also the logical propositions describing the network relations are continuous. Fuzzy logic is aimed to provide formal foundation to approximate reasoning, including common language (Zadeh, 1965; Dubois et al., 1997; Novak et al., 1999). It is characterized by a graded approach, so that the degree to which an object exhibits a property is specified by a characteristic function (specified below) with truth values ranging between completely false (0, inhibited, or unexpressed), to completely true (1, activated, or expressed). The theory satisfies the axiomatics as Boolean logic, with the exception of the principles of no-contradiction, and the excluded middle. The first one states that a proposition and its negation may not be simultaneously true; the second that, for any proposition, either that proposition is true or its negation is true. Fuzzy logic has been applied in a number of engineering applications, such as control systems or pattern recognition.

The Boolean network interactions may be extended to the fuzzy realm by means of the following rules:

p and q	$p \cdot q$
p or q	$p + q\text{-}p \cdot q$
not p	$1\text{-}p$

Since a proposition w and its negation 1-w may be simultaneously true, it follows that $w = 1\text{-}w$ is a valid statement with solution $w_{\text{thr}} = 1/2$ (Kosko, 1990). Thus, w_{thr} is a threshold value between falsity and truth or, equivalently, between inhibited and active, a result which we employ below.

The regulatory network consists of n interacting nodes with expression levels at a time t given by $q_i(t)$ $(i = 1,\ldots,v)$. The state of this node is regulated by its interaction with the rest of the network nodes, represented by a composite fuzzy proposition $w_i(q_i,\ldots,q_N)$ that summarizes experimental observations. Following similar lines as those employed in logistic inference, it may be shown that the expression level of w_i may be parameterized by a characteristic function with a logistic structure:

$$\Theta[w_i] = \frac{1}{1 + \exp[-b(w_i - w_{\text{thr}})]}$$

Here, the parameter b indicates the progression rate of w_i from false to true, gradual for small b, sharp for large b. Since we are interested in representing input functions with a differentiable step-like behavior we employ $b = 25$. The model predictions do not depend upon specific choices of b, as long as this parameter is large enough ($b \geq 10$) (**Supplementary Figure S1**).

Continuous Dynamical Model

The dynamic evolution of the expression level $q_i(t)$ is driven by the regulatory network interactions described by the membership function $\theta[w_i]$. The rate of change of $q_i(t)$ is thus determined by a set of ODEs (**Figure 1C** and **Supplementary Data Sheet S4**) of the form:

$$\frac{dq_i}{dt} = \Theta[w_i] - \alpha_i q_i$$

Here, α_i is the decay rate of the expression of node i, so that in absence of a regulatory interaction the node expression level suffers an exponential time decay at a rate α_i. In this paper we suppose that $\alpha_i = 1$ for all nodes, so that the stationary expression level of node i is merely given by the degree of truth of the fuzzy proposition w_i. The value of the parameter α_i does affect the transitions of the system. However, a sensitivity analysis of this parameter is beyond the scope of this paper and it merits a separate paper, as can be seen in Davila-Velderrain et al., 2015.

The resulting attractors of the dynamical system are presented in **Supplementary Data Sheet S4**. They may be obtained as asymptotic states of the network dynamics i.e., by considering the limit $t \to \infty$ of the solutions. They satisfy the steady-state condition $dq_i/dt = 0$, which leads to the expression

$$q_i^{ST} = \frac{1}{\alpha_i}\Theta[w_i(q_1^{ST}, \ldots, q_n^{ST})].$$

Although it is not the purpose of the present work, the continuous fuzzy description may be easily extended to a stochastic regime by adding a noise variable $\xi_i(t)$ (with appropriate statistical properties) at the right hand side of the ODE system (see Di Cara et al., 2007; Wittmann et al., 2009; Villarreal et al., 2012).

Polarization Analysis

The fuzzy logic model enabled evaluations of continuous alterations of the inputs (exogenous cytokines) and the intrinsic components (transcription factors, signaling pathways, and autocrine cytokines) of the network. To model polarization processes we studied the final steady states induced by stimulation associated to a specific cytokine environment on an initial Th0 state that corresponds to a CD4+ T cell under non-polarizing cytokine conditions. Dynamical simulations were performed for different sets of initial conditions and relative concentrations of microenvironmental cytokines to obtain the final steady states (**Figure 1D**). We considered that a node is actively expressed if its steady state value $q_i \geq 0.75$, unexpressed if $q_i \leq 0.25$, while intermediate values, $0.25 < q_i < 0.75$, correspond to a transition zone, with no definite expression. By using this criteria, it was stated that a steady state of the system corresponds to a CD4+ T cell subset if its corresponding transcription factors and cytokines are actively expressed, while states with null or low expression levels of all transcription factors were considered as Th0 (**Supplementary Data Sheet S5**).

Given the continuous nature of the regulatory network model presented here, it is impossible to determine all the possible steady states, since they are determined by an infinite set of initial conditions with expression values lying in the range [0,1]. We solved this problem by first verifying that the cell subtypes (or phenotypes) predicted by the discrete model are recovered in the continuous approach when the initial conditions are limited to the values 0 or 1; in that case, steady states stemming from the whole continuous range of initial conditions may be classified according to their similarity to cell types prognosticated by the Boolean model: Th0, Th1, Th2, Th17, Treg, Tfh, Th9, Tr1, and Th3 (**Supplementary Data Sheet S6**). It is understood that a continuous steady state is similar to Boolean state if its active nodes are coincident (with $q_i \geq 0.75$). Steady states with intermediate expression values were considered to be in a transition zone (t.z.) of phenotypic coexistence.

Plastic Transitions and Repatterning Analysis

In order to model plastic transitions, we considered a cell in an already partial or fully polarized state determined by different expression levels of the characteristic transcription factors and cytokines (**Figure 1D** and **Supplementary Data Sheet S3**), as defined before. In both kinds of simulations, we represented the effect of the microenvironment using a selected set of exogenous cytokines (**Table 1**) active at relative concentrations in the range $0 \geq q_i \geq 1$. Repatterning analyses were conducted numerically using an algorithm presented in Davila-Velderrain et al., 2015. A specific attractor was taken as an initial condition in an ODEs initial-value problem. For each active node in the attractor an ordered set of concentration values of exogenous cytokines was chosen, leaving constant the rest of system parameters. The ODEs were then solved numerically until reaching a steady state q_i^{ST}, each time using a slightly different exogenous cytokine concentration, and for all concentrations in the set. In order to identify bifurcating solutions of the ODE, a plot was generated for the total sum Q for the absolute value of the difference between the final and initial expression values of single-nodes

$$Q = \sum_{i=1}^{n} q_i^{ST}$$

as function of the varying expression value, as depicted in **Figures 2**, **3**. Phenotypic transitions are distinguished by the occurrence of notorious jumps of the parameter Q, denoted as distance in the bifurcation graphs. The former method was employed to investigate reported CD4+ T cell phenotypic transitions induced by environmental cytokines with high immunological and pathogenic relevance like Th1/Th2 and Treg/Th17. The code for all the simulation experiments performed in this work is available in **Supplementary Data Sheet S7**.

RESULTS

Effect of Exogenous Microenvironment on CD4+ T Cell Differentiation

To evaluate how altered concentrations of exogenous cytokines in the microenvironment shape $CD4^+$ T cell differentiation, we studied the activation process of a Th0 cell as a function of increasing concentrations of the exogenous cytokines and determined the final steady states (**Figure 3**). We found that the exogenous cytokines IL12e, IFNGe, IL4e, IL6e, IL21e, TGFBe, and IL10e induce the differentiation from a Th0 initial steady state toward Th1, Th2, Tfh, Th3, and Tr1 subsets, respectively. Experimentally, these cytokines have been described as sufficient to induce differentiation into their associated cell types and are part of the feedback loops with the characteristic transcription factors of such types (Zhu et al., 2010). On the other hand, our model predicts that Th17, Th9, and iTreg subsets are not induced by alterations in a single exogenous cytokine in the micro-environment. Th17 cells requires exogenous TGFβ in addition to IL6e/IL21e, Treg cells require constant IL-2 in the microenviroment in addition to TGFβ and Th9 cells are highly dependent on the presence of both IL-4 and TGFβ (Zhu et al., 2010; Schmitt et al., 2014).

The critical concentration required to induce a transition varied depending on the particular exogenous cytokine being modified. IL12e, IL6e, and IL21e required relatively small concentrations (0.2) to induce the differentiation from Th0 to Th1 and Tfh, respectively, while IL4e required a higher concentration (0.36) to induce the differentiation from Th0 to Th2. On the other hand, IL2e and IL27e alone were not able to induce transitions. We observed that IL2e induced the expression of high levels of IL2; however, we labeled the resulting cells as Th0, as IL-2 production by itself is not associated with a particular polarization subset.

It is also interesting to note that transitions among subsets have different patterns of sensitivity to exogenous cytokine concentrations. Most of the transitions from Th0 to other subsets were discontinuous; once a threshold concentration was achieved, the cell changed its expression pattern to a different one in an abrupt manner. An exception was observed when IL10 was used as an inducer. This cytokine caused a gradual transition from Th0 to Tr1; in this case, a continuous range of steady states was recovered in the transition zone between both subsets. These results predict that, for most of single cytokines, $CD4^+$ T cells should initiate differentiation once the threshold concentration has been reached, whereas these cells may display a range of sensitivities to altered concentrations of other cytokines in order to switch to a different state or phenotype.

$CD4^+$ T subsets such as Th9, Th17, and iTreg require particular combinations of cytokines to differentiate from naïve cells. In our model, we simulated the activation of a Th0 cell in the presence of different combinations and concentrations of the exogenous cytokines associated with the microenvironment (**Table 2** and **Figure 4**). In the case of requiring more than one exogenous cytokine, all the implicated nodes were set to the same value. Using this methodology, we were able to induce the differentiation from a Th0 steady state toward Th1, Th2, Th17, Th9, Tfh, iTreg, Th3, and Tr1 subsets by cytokine combinations that are in agreement with experimental data (Zhu et al., 2010; Crotty, 2014; DuPage and Bluestone, 2016).

The concentration required to induce polarization when using multiple cytokines varied depending on the $CD4^+$ T initial cell type. Under their combined action the individual concentrations are lower (**Figure 4**) than those required in the case of a single exogenous cytokine (**Figure 3**). This result suggests that the regulatory network mediates a synergistic effect of cytokines on $CD4^+$ T cell differentiation. For example, while a concentration of IL4e = 0.36 was necessary to induce the polarization toward Th2, a concentration of IL 2e and IL4e = 0.26 was sufficient to induce the same effect. Similarly, while a concentration of IL10e = 0.6 was necessary to induce the polarization toward Tr1, a concentration of IL10e and IL27e = 0.43 produced the same transition. Furthermore, autocrine IL10 achieved its maximum value with a lower concentration of exogenous cytokines when IL10e and IL27e act synergistically.

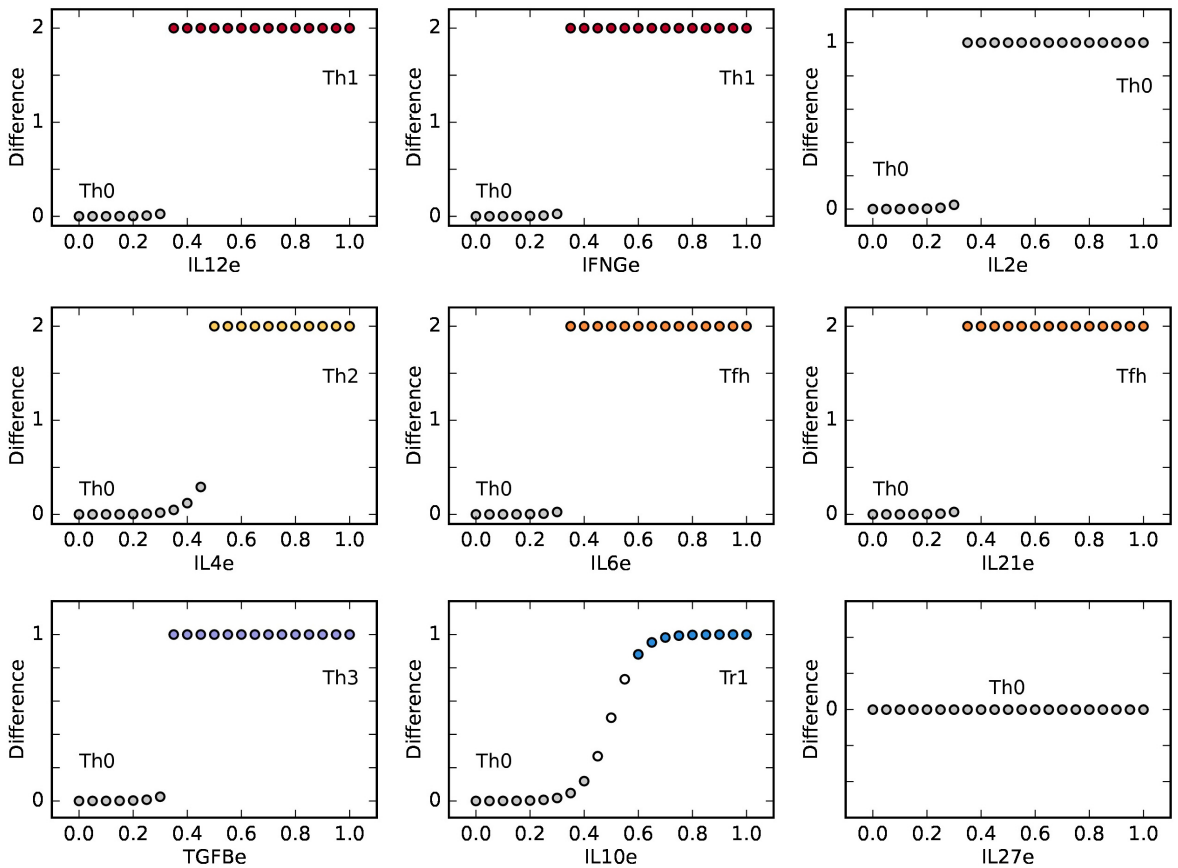


FIGURE 3 | CD4+ T cell fate as a function of the concentration of single exogenous cytokines: IL12, IFNG, IL2, IL4, IL6, Il21, TGFB, IL10, and IL27. From an initial state TH0, a CD4+ T cell may acquire diverse phenotypes on an abrupt or gradual transition, depending on critical concentrations of environmental cytokines. The plot shows the difference between the values of the initial Th0 state and the final steady state at different concentrations of exogenous cytokines. We observe that the presence of either IL12 or IFNg is sufficient for Th1 polarization, as well as IL4, is sufficient for TH2 polarization. On the other hand, IL2 alone does not lead to an effector phenotype. Similarly, the presence of either IL6 or IL21 alone is sufficient for Tfh induction, as is the case of TGFB and IL10, leading to Th3 and Tr1, respectively. IL27 alone does not lead to any fate transition in this model.

Figure 4 shows that differentiation processes in pro-Th1, pro-Th2, and pro-Tfh microenvironments were abrupt, while the transition in a pro-Tr1 environment was gradual. In a pro-Th17, pro-Th9, and pro-iTreg alterations in the micro-environments, including TGFβe, caused a small abrupt change followed by a gradual change in the expression levels of the components in the steady state configuration. In the pro-Th17 and pro-Th9 the model predicted an intermediate step before the final polarized state was achieved. In the pro-Th17 case, increasing cytokine levels induced an initial abrupt change toward a plateau zone corresponding to Tfh, followed by a transition to the Th17 steady state. A similar behavior was observed in the pro-Th9 microenvironment with a precursor TGFβ $^{+}$ (Th3) subset, followed by a final Th9 steady state. It is worth noting that TGFβ has a key role in the induction of the three types of CD4^{+} T cell types discussed here and it has complex interactions with other exogenous cytokines in their effects on cell plasticity (Eizenberg-Magar et al., 2017). These results illustrate that the continuous versión of our minimal CD4+ T cell differentiation model comprises a useful working hypothesis concerning the dynamic and complex mechanisms underlying how the microenvironment alters cell plasticity in response to TGFβ in the immune system.

In summary, the continuous model presented in this paper recovers CD4+ T cell plasticity responses to cytokine concentrations that have been documented experimentally and explains how such patterns of cell-type shifts depend on the initial CD4+ T cell type, being sometimes abrupt and others gradual. It also shows that cytokine combinations and, notably, the induction of different subsets under the action of different concentrations of the same cytokine combinations underlie different patterns of CD4+ T cell transitions.

Effects of the Exogenous and Endogenous Microenvironment on CD4^{+} T Cell Plasticity

We first focus on the transition between Th1 and Th2, that has been experimentally observed, particularly when these cells have recently differentiated, but not when they are fully

TABLE 2 | Exogenous cytokines in different environments included in the CD4+ T cell regulatory network.

Micro-environment	Active input nodes
pro-Th0	None
pro-Th1	IFNGe, IL12e
pro-Th2	IL2e, IL4e
pro-Th17	IL21e, TGFBe
proTh9	IL4e, TGFBe
proTfh,	IL21e
pro-iTreg	IL2e, TGFBe
pro-Tr1	IL10e, IL27e
pro-Th3	TGFBe

Active nodes refer to the same exogenous cytokines, whose concentrations were modified during the simulation, adopting values between 0 and 1.

polarized (Perez et al., 1995; Panzer et al., 2012). To study this process we considered the response of already differentiated Th1 and Th2 states, in response to variable concentrations of a defined cytokine for a particular subset, in combination with the opposing cytokine (IFNGe for Th2, and IL4e, for Th1), and then we used the model to predict the final steady state. **Figure 5** shows that when the initial configuration of the system corresponded to a highly polarized Th1 (TBET and IFNG = 1) or Th2 (GATA3 and IL4 = 1) states, for every combination of (exogenous) IL4e and IFNGe concentrations, the system remained in its original state even under high concentrations of all these cytokines. This, indicates that highly polarized Th1 or Th2 cells are not plastic. However, by considering initial lower concentrations of Th1 and Th2 transcription factors and cytokines, consistent with partial phenotype polarization, plastic transitions are predicted by the model. $CD4^+$ T cells require the production of high levels of autocrine IFNG and expression of TBET to maintain a Th1 phenotype. If the expression levels decrease, especially in the case of autocrine IFNG, Th1 cells are predicted to transit into Th2 cells. At the same time, the Th2 cells require the production of high levels of autocrine IL4 and expression of GATA3 to maintain a Th2 phenotype. If the initial expression levels decrease these cells are expected to transit to Th1 cells. At high initial levels of

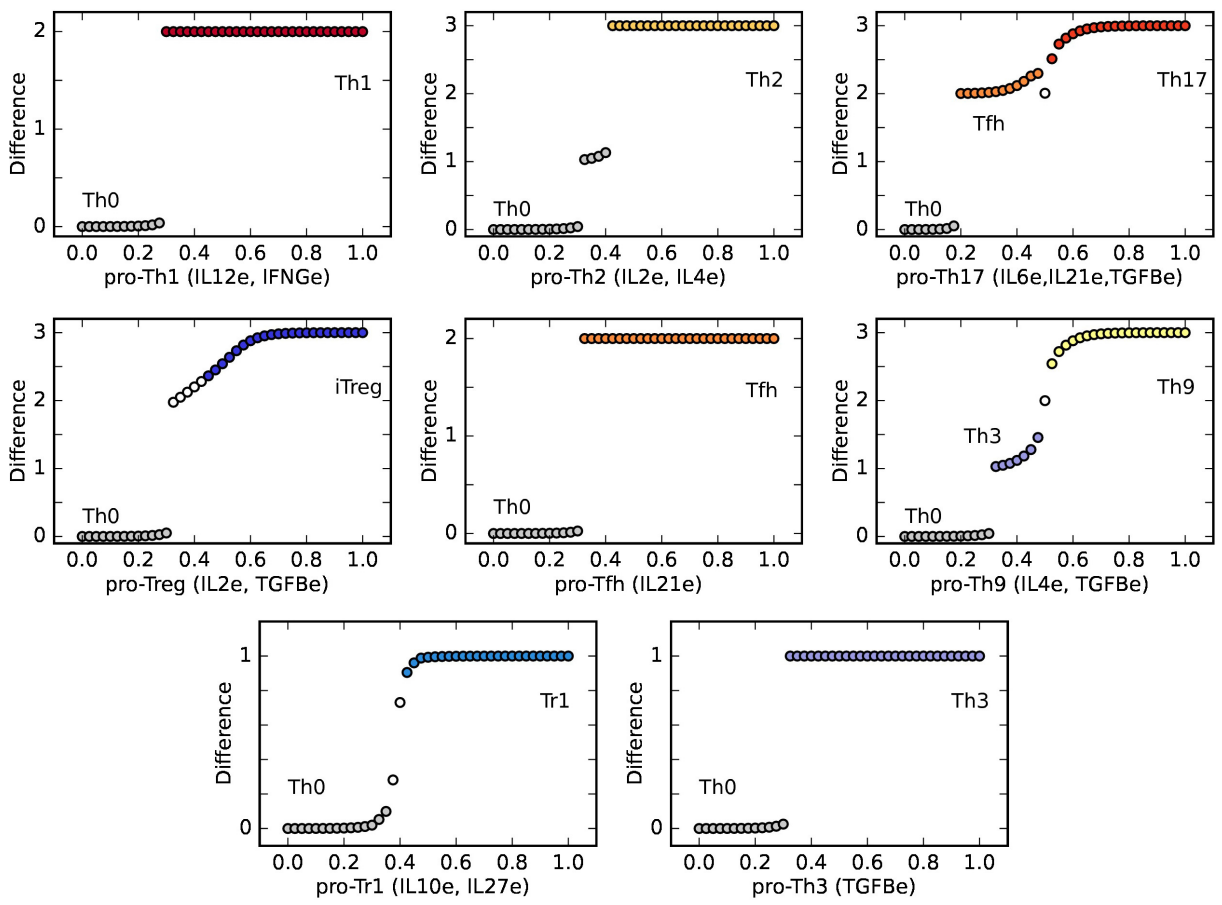


FIGURE 4 | T-CD4 cell fate as a function of exogenous cytokine concentrations define diverse phenotype-associated environments. From the Th0 initial state, a CD4+ T cell evolves to different phenotypes, depending on critical concentrations of environmental cytokines as shown in **Table 1**: Th1 (IFNG and IL12), Th2 (IL4, Il2), Th17 (Il21, TGFB), Treg (IL2, TGFB), Tfh (IL21), Th9 (IL4, TGFB), Tr1 (IL10, IL27), and Th3 (TGFB). The plot shows the difference between the values of the initial Th0 state and the final steady state at different concentrations of exogenous cytokines. The transition may be abrupt or gradual and, interestingly, may involve an intermediate state, as in the cases Th0 - > Tfh - > Th17 (C), and Th0 - > Th3 - > Th9 (F).

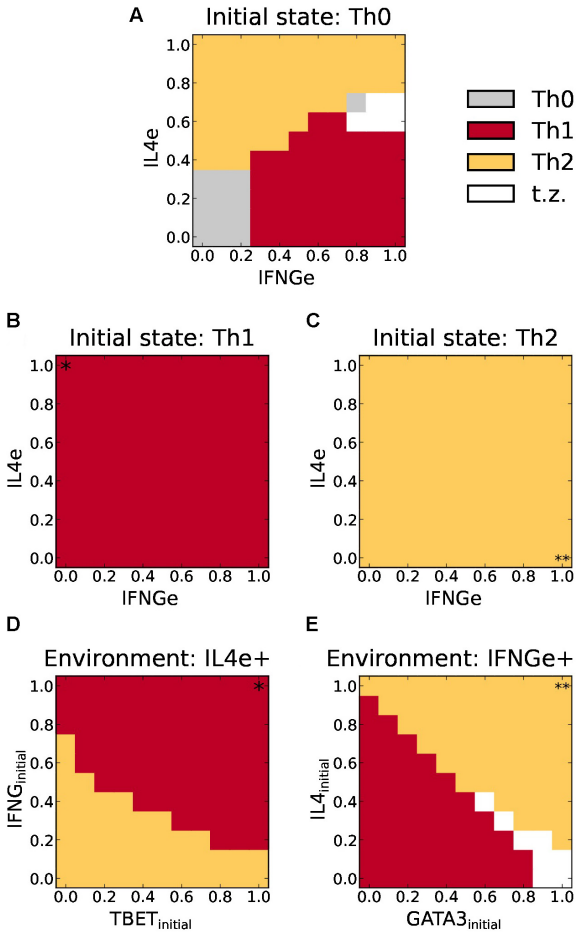


FIGURE 5 | Phenotype space diagrams for Th1 and Th2 polarization and plasticity as a function of the relative concentration of environmental IFNg and IL4, and expression of transcription factors. **(A)** Diagram for cell differentiation assuming an initial Th0 state. As the external concentration of IFNGe increases, the system develops an abrupt transition from Th0 to Th1. Similarly, an increase in external IL4e drives an abrupt transition from TH0 to Th2. For moderate concentrations of IFNGe and IL4e (< 0.8), we observe two wide zones of Th1 or Th2 prevalence with a sharp boundary, meaning that small variation of cytokines at these zones may change cell polarization. A transition zone with no defined polarization appears at higher concentrations of these cytokines (white and gray areas). **(B,C)** Plasticity diagrams assuming full Th1 **(B)** or Th2 **(C)** polarized states (i.e., induced by INFg = 1 and IL-4 = 1 in diagram **A**, respectively). No phenotypic transitions are observed under variable concentrations of environmental IL4e and autocrine IFNGe. **(D)** Plasticity diagram of Th1 cells assuming an environmental concentration of IL4e = 1. Cells require the production of initial high levels of autocrine IFNG and expression of TBET to maintain a Th1 phenotype. If the initial expression levels decrease, especially in the case of autocrine IFNG, it will transit into a Th2 cell. **(E)** Plasticity diagram of Th2 cells assuming an environmental concentration of IFNGe = 1. The cell requires the production of high levels of autocrine IL4 and expression of GATA3 to maintain a Th2 phenotype. If the initial expression levels decrease it will transit into a Th1 cell. At high expression levels of initial GATA3 and low initial IL4, there exists a transition zone where the cell cannot be classified.

GATA3 and low IL4, a transition zone at which cells display mixed characteristics is predicted. These results show that plasticity between the Th1 and Th2 subsets depends not only on the microenvironment cytokines, but also on the intracellular state.

The transition between Th17 and iTreg, has been extensively investigated experimentally (Xu et al., 2007; Wei et al., 2008; Lee et al., 2009a,b; Littman and Rudensky, 2010; Kleinewietfeld and Hafler, 2013; Noack and Miossec, 2014) and is particularly important for some pathological conditions, such as chronic inflammation. To study this process we considered fully differentiated Th17 (RORGT and IL21 = 1) and iTreg cells (FOXP3 and TGFB = 1) under the presence of different concentrations of the exogenous cytokines, IL2e, IL21e, and TGFBe. In the case of Th17 cells, they remained in a Th17 phenotype at a high concentration of TGFBe, while they switched toward Tfh for lower concentrations of TGFBe (< 0.6). Some experiments have reported that induction of Th17 require exogenous TGFB (Veldhoen et al., 2006), but it is uncertain if the transition toward Tfh associated to low TGFB levels will occur in all cases. On the other hand, iTreg cells remain stable under high concentrations of IL2e, while they transit toward Th17, Tfh, or Th3 at low concentrations of IL2e (< 0.65) (**Figure 6**). These results show that plastic transitions between subsets are not symmetrical, and depend on the previous polarization state of the cell.

DISCUSSION

Our simulations show contrasting differentiation patterns of CD4+ T cells under different concentrations and combinations of exogenous cytokines, highlighting the importance of synergy and competing interactions among microenvironment components and CD4+ T cell network components to induce different patterns of CD4+ T cell plasticity. We also showed that plasticity between the Th1/Th2 and iTreg/Th17 subsets depends on varying the concentration of microenvironment cytokines and the expression level of intracellular transcription factors and autocrine cytokines depending on the initial cell type.

The model predicts both abrupt and gradual transitions between cell types. In abrupt transitions, there is a sudden change from an initial to a final steady state or cell type, once the concentration of exogenous cytokines exceeds a threshold value. This behavior suggests that the transition between stable cell phenotypes is energetically favorable once the threshold value has been achieved. In this process, exogenous cytokines provide the initial stimulus to promote the expression of both transcription factors and autocrine cytokines characteristic of a cell type that is different to the original one, while positive feedback loops greatly increase their polarization efficiency.

In contrast, in gradual transitions, steady states that express intermediate levels of transcription factors and autocrine cytokines appear. In these steady states, a clear-cut threshold between the two expression patterns is not observed, so they cannot be easily classified into one subset or another, signaling the manifestation of partially polarized states. The heterogeneity of CD4+ T cells has been well-documented (Murphy and Stockinger, 2010; DuPage and Bluestone, 2016; Eizenberg-Magar et al., 2017), and could be the result of regulatory circuits capable

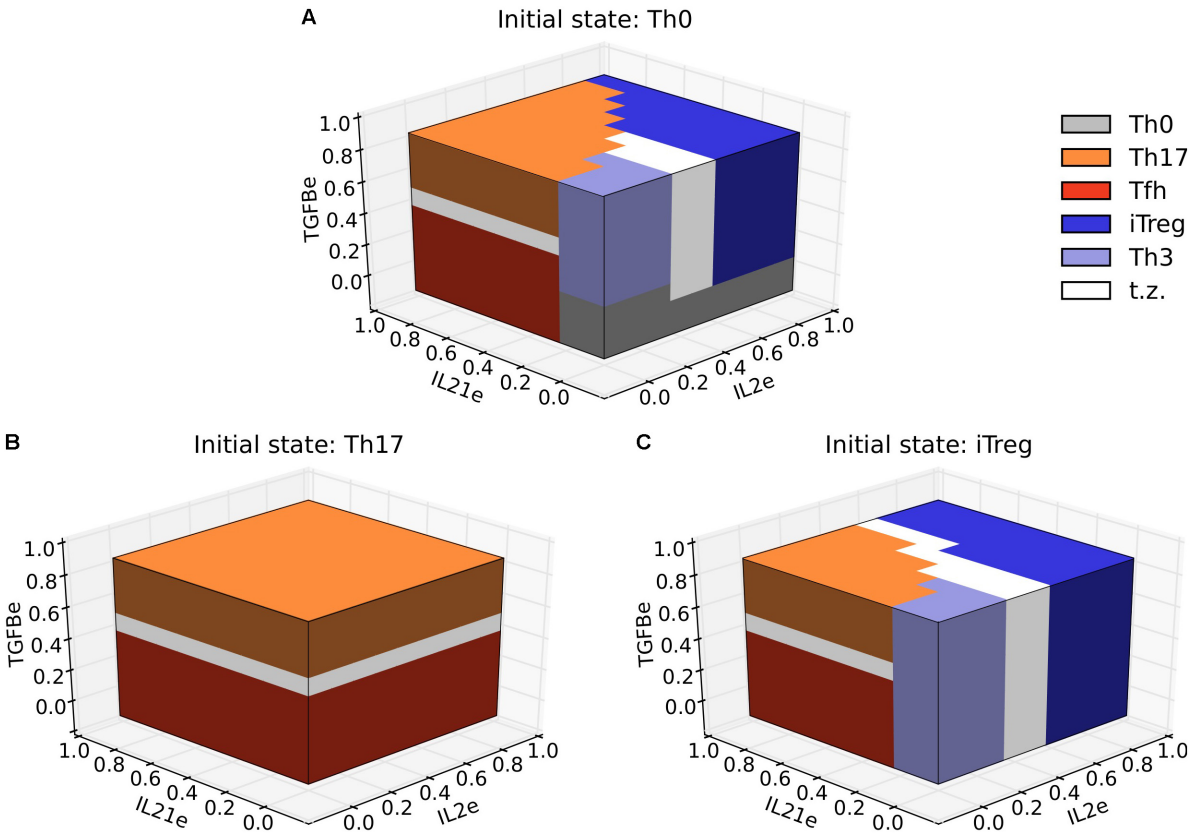


FIGURE 6 | Three-dimensional phenotype space diagrams for Th17 and iTreg polarization and plasticity as a function of the relative concentrations of IL2, IL21, and TGFB in the microenvironment. In the differentiation diagram **(A)** we observe alternative phenotypic regions defined by relative concentrations of environmental cytokines. The regions may be either separated by a sharp boundary or by a more gradual transition zone (labeled in white). The plasticity diagram **(B)** indicates a polarized behavior for Th17 versus Tfh phenotype determined by a high or low concentration of external TGFB. A richer behavior ensues when the initial state is Treg, as shown in the plasticity diagram **(C)**. We observe a similar structure as that depicted in **A**, except that the Th0 zone is absent.

of generating a range of cells that express intermediate levels of specific molecules that can stably coexist or change from one another under certain conditions. It is important to notice that every gradual transition involves regulatory circuits with central nodes which display feedback interactions. Such feedback loops render stability to the initial polarization state so that its intrinsic cytokine production and transcription factor expression should gradually decrease under changing microenvironmental conditions. We observed this behavior especially in response to changes in the concentration of IL-10 and TGFβ. IL-10 is a regulatory cytokine produced by multiple CD4+ T subsets (Howes et al., 2014; Gagliani et al., 2015). TGFβ may display both regulatory and inflammatory effects and it is implied in the differentiation of multiple subsets like Th17, iTreg, and Th9 (Chen et al., 2003; Veldhoen et al., 2006; Davidson et al., 2007; Kaplan, 2013). It is conceivable that gradual transitions and generation of intermediate polarization states reflect the intricate regulatory signaling effects of TGFβ and of IL-21, and are probably responsible for tuning the effects of different conditions in the immune response (Grossman and Paul, 2015).

The model also captures some cases where there is an abrupt transition followed by a gradual transition in polarization processes. Such is the case of the Th0-Tfh-Th17, the Th0-Th3(TGFB+)-Th9 and the Th0-iTreg transitions. Interestingly, in all these cases TGFβ is present in the micro-environment. This indicates that the concentration of TGFβ may modulate the immune response in complex ways. These interesting results suggest a system-level explanation of previous experimental results. For example, it is known that TGFβ regulates Th17 cells in a differential way depending on the concentration and combinations of cytokines in the microenvironment (Yang et al., 2008). Furthermore, consistent with our simulations, it is known that the TGFβ signaling pathway is highly modulated (Attisano and Wrana, 2002; Travis and Sheppard, 2014). Our model also predicts that TGFβ may induce distinct subsets at different concentrations, in particular, Tfh, Th9, iTreg, and Th3. A careful analysis of this kind of regulatory circuits will shed light on the specific mechanisms defining transcriptional programs that lead to cell heterogeneity. Understanding the interactions underlying the dynamical behavior of T helper cells may help elucidate the regulatory role of this important molecule in the immune response.

The model presented in this paper also highlights the cooperation among different exogenous cytokines during

differentiation. Th17, iTreg, and Th9 subsets require TGFβ in combination with IL-6/IL-21, IL-2, and IL-4 to differentiate, respectively, in agreement with experimental data (Chen et al., 2003; Veldhoen et al., 2006; Davidson et al., 2007; Kaplan, 2013). In other cases, the effect of a single cytokine is sufficient to induce polarization, but the synergy with other cytokines lowers the threshold concentration necessary to induce polarization. In this way, the model allows us to study and predict synergic relations among cytokines in CD4+ T cell differentiation.

As mentioned above, we also use the model to study the effect of opposing cytokines in differentiation and plasticity of Th1/Th2 and Th17/iTreg subsets. The Th1 and Th2 cells are highly stable, and the transition between them is hard to achieve experimentally (Perez et al., 1995; Murphy et al., 1996; Hegazy et al., 2010). Coincidently our model shows that, once these types have achieved a stable state, Th1 and Th2 are robust to changes in their microenvironment. This behavior seems consistent with a particularly robust interaction circuit, defined by coupled regulatory switching modules between mutually inhibitory nodes with negative feedbacks, each node defining an alternative regulatory route. However, partially polarized cells can transit to the other cell types when they are subject to an opposing cytokine (IL-4 in the case of Th1 or IFNγ in the case of Th2). In conclusion, our model provides a system-level mechanistic explanation to these complex behaviors of Th1 and Th2 cells.

The model also recovers the spontaneous transition of iTreg into Th17 in the presence of IL-21 or the closely similar IL-6 (here considered as equivalents) (Xu et al., 2007) at low concentrations of IL-2. The plasticity of this transition is not symmetrical, as changes in the microenvironment are not enough for Th17 to transit toward iTreg. For such transition, it is also necessary to alter the internal state of the cell, changing the expression levels of key transcription factors, as it has been shown in experimental studies (Michalek et al., 2011; Berod et al., 2014; Gagliani et al., 2015). These results seem to imply that the basin of attraction of iTreg is shallower than that of Th17. This could be the result of the different regulatory circuits implied in the differentiation of each cell type, since while both depend on TGFβ, iTreg both require and inhibit the production of IL-2 (Fontenot et al., 2003; Pandiyan et al., 2007), restricting the stability of these cells.

The model and simulations presented here are able to describe cell type transitions and the recovered patterns do not rely upon specific parameter estimates, but rather on the network structure and overall dynamic behavior. However, the exact transition points may change depending on the precise concentrations and parameters of the biological system (Eizenberg-Magar et al., 2017). Given the relative nature of the semi-quantitative variations introduced in the model, we should be cautious in providing precise quantitative predictions concerning the sensitivity of the different subsets under real experimental conditions. Theoretical models like the one presented here provide an ideal tool to integrate recent advances in experimental knowledge and provide a system-level mechanistic explanation for observed behaviors in experiments, and also to provide informed predictions for future experiments. Hence, the feedback between experimental and theoretical research is necessary to understand the rich behavior of CD4+ T cells and the immunological system.

CONCLUSION

The continuous model with fuzzy logic interaction rules, presented in this paper, recovers CD4+ T cell plasticity responses to cytokine concentrations that have been documented experimentally and explains how such patterns of cell-type shifts results from feedback between initial T cell type and the microenvironment, being sometimes abrupt and others gradual. The simulations show how different cytokine combinations and, notably, the induction of different subsets under the action of different concentrations of the same cytokine combinations underlie different patterns of T cell transitions. The semi-quantitative nature of the model allows predictions that do not depend on specific parameters for which we are still lacking experimental support. This model may contribute to the study of immunological diseases where cellular plasticity is a key, such as cancer, and autoimmune diseases like type 1 diabetes, multiple sclerosis, or juvenile arthritis (DuPage and Bluestone, 2016).

AUTHOR CONTRIBUTIONS

EA-B and CVL conceived, planned, and coordinated the study. CVL and MM-S established the continuous model and performed simulations and calculations. LH contributed with her expertise on T cell signaling and immunological consequences. All authors participated in the interpretation, analyses of results, and wrote the paper.

FUNDING

EA-B and MM-S received funding from CONACYT: 240180, 180380, 2015-01-687, and UNAM-DGAPA-PAPIIT: IN211516, ININ208517, IN205517, and IN204217. CVL received funding from CONACYT: 180380. LH received funding from CONACYT: CB2014/238931 and UNAM-PAPIIT IN211716.

ACKNOWLEDGMENTS

We acknowledge Diana Romos for her support with logistical tasks. We thank Jose Davila-Velderrain and Juan Arias del Angel for providing code for this project.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphys.2018.00877/full#supplementary-material

FIGURE S1 | Sensitivity analysis of the parameter *b*. Effect of various values of *b* (5, 10, 25, and 50) in abrupt (IL2e), gradual (IL10e), and mixed (IL6e + IL21e + TGFBe) transitions. The model predictions do not depend upon the specific choice of b if this parameter is large enough ($b \geq 10$).

DATA SHEET S1 | References of the CD4+ T cell regulatory network.

DATA SHEET S2 | Boolean rules of the CD4+ T cell regulatory network.

DATA SHEET S3 | Boolean attractors of the CD4+ T cell regulatory network.

DATA SHEET S4 | Ordinary differential model equations of the CD4+ T cell regulatory network.

DATA SHEET S5 | Boolean rules for labelling the attractors of the CD4+ T cell regulatory network.

DATA SHEET S6 | Continuous attractors of the CD4+ T cell regulatory network.

DATA SHEET S7 | Code and simulations of the CD4+ T cell regulatory network.

REFERENCES

Abou-Jaoudé, W., Monteiro, P. T., Naldi, A., Grandclaudon, M., Soumelis, V., Chaouiya, C., et al. (2014). Model checking to assess T-helper cell plasticity. *Front. Bioeng. Biotechnol.* 2:86. doi: 10.3389/fbioe.2014.00086

Albert, R., and Thakar, J. (2014). Boolean modeling: a logic-based dynamic approach for understanding signaling and regulatory networks and for making useful predictions. *Wiley Interdiscip. Rev. Syst. Biol. Med.* 6, 353–369. doi: 10.1002/wsbm.1273

Alvarez-Buylla, E. R., Davila-Velderrain, J., and Martinez-Garcia, J. C. (2016). Systems biology approaches to development beyond bioinformatics: nonlinear mechanistic models using plant systems. *BioScience* 66, 371–383. doi: 10.1093/biosci/biw027

Ansel, K. M., Djuretic, I., Tanasa, B., and Rao, A. (2006). Regulation of TH2 Differentiation and il4 locus accessibility. *Annu. Rev. Immunol.* 24, 607–656. doi: 10.1146/annurev.immunol.23.021704

Assenmacher, M., Schmitz, J., and Radbruch, A. (1994). Flow cytometric determination of cytokines in activated murine T helper lymphocytes: expression of interleukin-10 in interferon and in interleukin 4 expressing cells. *Eur. J. Immunol.* 24, 1097–1101. doi: 10.1002/eji.1830240513

Attisano, L., and Wrana, J. L. (2002). Signal transduction by the TGF-beta superfamily. *Science* 296, 1646–1647. doi: 10.1126/science.1071809

Awasthi, A., Carrier, Y., Peron, J. P. S., Bettelli, E., Kamanaka, M., Flavell, R. A., et al. (2007). A dominant function for interleukin 27 in generating interleukin 10-producing anti-inflammatory T cells. *Nat. Immunol.* 8, 1380–1389. doi: 10.1038/ni1541

Azpeitia, E., Benitez, M., Padilla-Longoria, P., Espinosa-Soto, C., and Alvarez-Buylla, E. R. (2011). Dynamic network-based epistasis analysis: boolean examples. *Front. Plant Sci.* 2:92. doi: 10.3389/fpls.2011.00092

Azpeitia, E., Davila-Velderrain, J., Villarreal, C., and Alvarez-Buylla, E. R. (2014). "Gene regulatory network models for floral organ determination," in *Methods in Molecular Biology*, eds J. L. Riechmann and F. Wellmer (New York, NY: Springer).

Barberis, M., Helikar, T., and Verbruggen, P. (2018). Simulation of stimulation: cytokine dosage and cell cycle crosstalk driving timing-dependent T Cell differentiation. *Front. Physiol.* 9:879. doi: 10.3389/fphys.2018.00879

Berod, L., Friedrich, C., Nandan, A., Freitag, J., Hagemann, S., Harmrolfs, K., et al. (2014). *De novo* fatty acid synthesis controls the fate between regulatory T and T helper 17 cells. *Nat. Med.* 20, 1327–1333. doi: 10.1038/nm.3704

Bornholdt, S. (2008). Boolean network models of cellular regulation: prospects and limitations. *J. R. Soc. Interface* 5(Suppl. 1), S85–S94. doi: 10.1098/rsif.2008.0132.focus

Bucy, R. P., Panoskaltsis-Mortari, A., Huang, G., Li, J., Karr, L., Ross, M., et al. (1994). Heterogeneity of single cell cytokine gene expression in clonal T cell populations. *J. Exp. Med.* 180, 1251–1262. doi: 10.1084/jem.180.4.1251

Carbo, A., Hontecillas, R., Kronsteiner, B., Viladomiu, M., Pedragosa, M., Lu, P., et al. (2013). Systems modeling of molecular mechanisms controlling cytokine-driven CD4+ T cell differentiation and phenotype plasticity. *PLos Comput. Biol.* 9:e1003027. doi: 10.1371/journal.pcbi.1003027

Chang, J. T., Palanivel, V. R., Kinjyo, I., Schambach, F., Intlekofer, A. M., Banerjee, A., et al. (2007). Asymmetric T lymphocyte division in the initiation of adaptive immune responses. *Science* 315, 1687–1691. doi: 10.1126/science.1139393

Chen, W., Jin, W., Hardegen, N., Lei, K.-J., Li, L., Marinos, N., et al. (2003). Conversion of Peripheral CD4 + CD25- Naive T Cells to CD4 + CD25 + regulatory T cells by TGF-β induction of transcription factor Foxp3. *J. Exp. Med.* 198, 1875–1886. doi: 10.1084/jem.20030152

Cortes, Y., Lotto, R. B., Malkin, D., Gerardo, J., Espinosa-Soto, C., Hartasa, D. A., et al. (2008). Floral morphogenesis: stochastic explorations of a gene network epigenetic landscape. *PLos One* 3:e3626. doi: 10.1371/journal.pone.0003626

Cote-Sierra, J., Foucras, G., Guo, L., Chiodetti, L., Young, H. A., Hu-Li, J., et al. (2004). Interleukin 2 plays a central role in Th2 differentiation. *Proc. Natl. Acad. Sci. U.S.A.* 101, 3880–3885. doi: 10.1073/pnas.0400339101

Crotty, S. (2014). T follicular helper cell differentiation, function, and roles in disease. *Immunity* 41, 529–542. doi: 10.1016/j.immuni.2014.10.004

Davidson, T. S., DiPaolo, R. J., Andersson, J., and Shevach, E. M. (2007). Cutting edge: IL-2 is essential for TGFB mediated induction of Foxp3 + T regulatory cells. *J. Immunol.* 178, 4022–4026. doi: 10.4049/jimmunol.178.7.4022

Davila-Velderraín, J., Martínez-García, J. C., and Álvarez-Buylla, E. R. (2017). *Boolean Dynamic Modeling Approaches to Study Plant Gene Regulatory Networks: Integration, Validation, and Prediction*. New York, NY: Humana Press.

Davila-Velderrain, J., Villarreal, C., and Alvarez-Buylla, E. R. (2015). Reshaping the epigenetic landscape during early flower development: induction of attractor transitions by relative differences in gene decay rates. *BMC Syst. Biol.* 9:20. doi: 10.1186/s12918-015-0166-y

Di Cara, A., Garg, A., De Micheli, G., Xenarios, I., and Mendoza, L. (2007). Dynamic simulation of regulatory networks using SQUAD. *BMC Bioinformatics* 8:462. doi: 10.1186/1471-2105-8-462

Dubois, D., Moral, S., and Prade, H. (1997). A semantics for possibility theory based on likelihoods. *J. Math. Anal. Appl.* 205, 359–380. doi: 10.1006/jmaa.1997.5193

DuPage, M., and Bluestone, J. A. (2016). Harnessing the plasticity of CD4(+) T cells to treat immune-mediated disease. *Nat. Rev. Immunol.* 16, 149–163. doi: 10.1038/nri.2015.18

Duque, G. A., and Descoteaux, A. (2014). Macrophage cytokines: involvement in immunity and infectious diseases. *Front. Immunol.* 5:491. doi: 10.3389/fimmu.2014.00491

Eizenberg-Magar, I., Rimer, J., Zaretsky, I., Lara-Astiaso, D., Reich-Zeliger, S., and Friedman, N. (2017). Diverse continuum of CD4 + T-cell states is determined by hierarchical additive integration of cytokine signals. *Proc. Natl. Acad. Sci. U.S.A.* 114, E6447–E6456. doi: 10.1073/pnas.1615590114

Fontenot, J. D., Gavin, M. A., and Rudensky, A. Y. (2003). Foxp3 programs the development and function of CD4 + CD25 + regulatory T cells. *Nat. Immunol.* 4, 330–336. doi: 10.1038/ni904

Gagliani, N., Vesely, M. C. A., Iseppon, A., Brockmann, L., Xu, H., Palm, N. W., et al. (2015). Th17 cells transdifferentiate into regulatory T cells during resolution of inflammation. *Nature* 523, 221–225. doi: 10.1038/nature14452

Gol-Ara, M., Jadidi-Niaragh, F., Sadria, R., Azizi, G., and Mirshafiey, A. (2012). The role of different subsets of regulatory t cells in immunopathogenesis of rheumatoid. *Arthritis* 2012:805875. doi: 10.1155/2012/805875

Grossman, Z., and Paul, W. E. (2015). Dynamic tuning of lymphocytes: physiological basis, mechanisms, and function. *Annu. Rev. Immunol.* 33, 677–713. doi: 10.1146/annurev-immunol-032712-100027

Haken, H. (1977). *Synergetics*. Berlin: Springer.

Hegazy, A. N., Peine, M., Helmstetter, C., Panse, I., Frohlich, A., Bergthaler, A., et al. (2010). Interferons direct Th2 cell reprogramming to generate a stable GATA-3^{+}T-bet^{+} cell subset with combined Th2 and Th1 cell functions. *Immunity* 32, 116–128. doi: 10.1016/j.immuni.2009.12.004

Hori, S., Nomura, T., and Sakaguchi, S. (2003). Control of regulatory T cell development by the transcription factor Foxp3. *Science* 299, 1057–1061. doi: 10.1126/science.1079490

Howes, A., Stimpson, P., Redford, P., Gabrysova, L., and O'Garra, A. (2014). Interleukin-10: cytokines in anti-inflammation and tolerance. *Cytokine Front.* 6, 327–352. doi: 10.1007/978-4-431-54442-5

Hsieh, C. S., Macatonia, S. E., Tripp, C. S., Wolf, S. F., O'Garra, A., and Murphy, K. M. (1993). Development of TH1 CD4 + T cells through IL-12 produced by Listeria-induced macrophages. *Science* 260, 547–549. doi: 10.1126/science.8097338

Ivanov, I. I., McKenzie, B. S., Zhou, L., Tadokoro, C. E., Lepelley, A., Lafaille, J. J., et al. (2006). The orphan nuclear receptor RORγt directs the differentiation program of proinflammatory IL-17 + T helper cells. *Cell* 126, 1121–1133. doi: 10.1016/j.cell.2006.07.035

Johnston, R. J., Poholek, A. C., DiToro, D., Yusuf, I., Eto, D., Barnett, B., et al. (2009). Bcl6 and Blimp-1 are reciprocal and antagonistic regulators of T follicular helper cell differentiation. *Science* 325, 1006–1010. doi: 10.1126/science.1175870

Kaplan, M. H. (2013). Th9 cells: differentiation and disease. *Immunol. Rev.* 252, 104–115. doi: 10.1111/imr.12028

Kauffman, S. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.* 22, 437–467. doi: 10.1016/0022-5193(69)90015-0

Kelso, A., Groves, P., Ramm, L., and Doyle, A. G. (1999). Single-cell analysis by RT-PCR reveals differential expression of multiple type 1 and 2 cytokine genes among cells within polarized CD4 + T cell populations. *Int. Immunol.* 11, 617–621. doi: 10.1093/intimm/11.4.617

Kleinewietfeld, M., and Hafler, D. A. (2013). The plasticity of human Treg and Th17 cells and its role in autoimmunity. *Semin. Immunol.* 25, 305–312. doi: 10.1016/j.smim.2013.10.009

Koch, M. A., Tucker-Heard, G., Perdue, N. R., Killebrew, J. R., Urdahl, K. B., Campbell, D. J., et al. (2009). The transcription factor T-bet controls regulatory T cell homeostasis and function during type 1 inflammation. *Nat. Immunol.* 10, 595–602. doi: 10.1038/ni.1731

Korn, T., Bettelli, E., Oukka, M., and Kuchroo, V. K. (2009). IL-17 and Th17 Cells. *Annu. Rev. Immunol.* 27, 485–517. doi: 10.1146/annurev.immunol.021908.132710

Kosko, B. (1990). Fuzziness vs. Probability. *Int. J. Gen. Syst.* 17, 211–240. doi: 10.1080/03081079008935108

Le Gros, G., Ben-Sasson, S. Z., Seder, R., Finkelman, F. D., Paul, W. E., Le Gros, G., et al. (1990). Generation of interleukin 4 (IL-4)-producing cells in vivo and in vitro: IL-2 and IL-4 are required for in vitro generation of IL-4-producing cells. *J. Exp. Med.* 172, 921–929. doi: 10.1017/CBO9781107415324.004

Lee, Y. K., Mukasa, R., Hatton, R. D., and Weaver, C. T. (2009a). Developmental plasticity of Th17 and Treg cells. *Curr. Opin. Immunol.* 21, 274–280. doi: 10.1016/j.coi.2009.05.021

Lee, Y. K., Turner, H., Maynard, C. L., Oliver, J. R., Chen, D., Elson, C. O., et al. (2009b).). Late developmental plasticity in the T helper 17 lineage. *Immunity* 30, 92–107. doi: 10.1016/j.immuni.2008.11.005

Littman, D. R., and Rudensky, A. Y. (2010). Th17 and regulatory T cells in mediating and restraining inflammation. *Cell* 140, 845–858. doi: 10.1016/j.cell.2010.02.021

Lu, Y., Hong, S., Li, H., Park, J., Hong, B., Wang, L., et al. (2012). Th9 cells promote antitumor immune responses in vivo. *J. Clin. Invest.* 122, 4160–4171. doi: 10.1172/JCI65459

Magombedze, G., Reddy, P. B. J., Eda, S., and Ganusov, V. V. (2013). Cellular and population plasticity of helper $CD4^+$ T cell responses. *Front. Physiol.* 4:206. doi: 10.3389/fphys.2013.00206

Martinez-Sanchez, M. E., Mendoza, L., Villarreal, C., and Alvarez-Buylla, E. R. (2015). A minimal regulatory network of extrinsic and intrinsic factors recovers observed patterns of CD4 + T cell differentiation and plasticity. *PLos Comput. Biol.* 11:e1004324. doi: 10.1371/journal.pcbi.1004324

Martínez-Sosa, P., and Mendoza, L. (2013). The regulatory network that controls the differentiation of T lymphocytes. *BioSystems* 113, 96–103. doi: 10.1016/j.biosystems.2013.05.007

Mendoza, L. (2006). A network model for the control of the differentiation process in Th cells. *BioSystems* 84, 101–114. doi: 10.1016/j.biosystems.2005.10.004

Mendoza, L., Thieffry, D., and Alvarez-Buylla, E. R. (1999). Genetic control of flower morphogenesis in *Arabidopsis thaliana*: a logical analysis. *Bioinformatics* 15, 593–606. doi: 10.1093/bioinformatics/15.7.593

Michalek, R. D., Gerriets, V. A., Jacobs, S. R., Macintyre, A. N., MacIver, N. J., Mason, E. F., et al. (2011). Cutting edge: distinct glycolytic and lipid oxidative metabolic programs are essential for effector and regulatory CD4 + T cell subsets. *J. Immunol.* 186, 3299–3303. doi: 10.4049/jimmunol.1003613

Moore, K. W., de Waal Malefyt, R., Coffman, R. L., and O'Garra, A. (2001). Interleukin -10 and the I Nterleukin -10 R Eceptor. *Annu. Rev. Immunol.* 19, 683–765. doi: 10.1146/annurev.immunol.19.1.683

Murphy, E., Shibuya, K., Hosken, N., Openshaw, P., Maino, V., Davis, K., et al. (1996). Reversibility of T helper 1 and 2 populations is lost after long-term stimulation. *J. Exp. Med.* 183, 901–913. doi: 10.1084/jem.183.3.901

Murphy, K. M., and Stockinger, B. (2010). Effector T cell plasticity: flexibility in the face of changing circumstances. *Nat. Immunol.* 76, 211–220. doi: 10.1007/s11103-011-9767-z.Plastid

Murugaiyan, G., Mittal, A., Lopez-Diego, R., Maier, L. M., Anderson, D. E., and Weiner, H. L. (2009). IL-27 is a key regulator of IL-10 and IL-17 production by human CD4 + T cells. *J. Immunol.* 183, 2435–2443. doi: 10.4049/jimmunol.0900568

Naldi, A., Carneiro, J., Chaouiya, C., and Thieffry, D. (2010). Diversity and plasticity of Th cell types predicted from regulatory network modelling. *PLos Comput. Biol.* 6:e1000912. doi: 10.1371/journal.pcbi.1000912

Naldi, A., Monteiro, P. T., Mussel, C., Kestler, H. A., Thieffry, D., Xenarios, I., et al. (2015). Cooperative ¨ development of logical modelling standards and tools with CoLoMoTo. *Bioinformatics* 31, 1154–1159. doi: 10.1093/bioinformatics/btv013

Noack, M., and Miossec, P. (2014). Th17 and regulatory T cell balance in autoimmune and inflammatory diseases. *Autoimmun. Rev.* 13, 668–677. doi: 10.1016/j.autrev.2013.12.004

Novak, V., Perfiljeva, I., and Mockor, J. (1999). *Mathematical Principles of Fuzzy Logic*. Boston, MA: Kluwer Academic Publishers. doi: 10.1007/978-1-4615-5217-8

Nurieva, R. I., Chung, Y., Martinez, G. J., Yang, X. O., Tanaka, S., Matskevitch, T. D., et al. (2009). Bcl6 mediates the development of T follicular helper cells. *Science* 325, 1001–1005. doi: 10.1126/science.1176676

Openshaw, P., Murphy, E. E., Hosken, N. A., Maino, V., Davis, K., Murphy, K., et al. (1995). Heterogeneity of intracellular cytokine synthesis at the single-cell level in polarized T helper 1 and T helper 2 populations. *J. Exp. Med.* 182, 1357–1367. doi: 10.1084/jem.182.5.1357

Pandiyan, P., Zheng, L., Ishihara, S., Reed, J., and Lenardo, M. J. (2007). $CD4^+CD25^+Foxp3^+$ regulatory T cells induce cytokine deprivation-mediated apoptosis of effector $CD4^+$ T cells. *Nat. Immunol.* 8, 1353–1362. doi: 10.1038/ni1536

Panzer, M., Sitte, S., Wirth, S., Drexler, I., Sparwasser, T., and Voehringer, D. (2012). Rapid in vivo conversion of effector T cells into Th2 cells during helminth infection. *J. Immunol.* 188, 615–623. doi: 10.4049/jimmunol.1101164

Perez, V. L., Lederer, J. A., Lichtman, A. H., and Abbas, A. K. (1995). Stability of Th1 and Th2 populations. *Int. Immunol.* 7, 869–875. doi: 10.1093/intimm/7.5.869

Perez-Ruiz, R. V., García-Ponce, B., Marsch-Martínez, N., Ugartechea-Chirino, Y., Villajuana-Bonequi, M., De Folter, S., et al. (2015). XAANTAL2 (AGL14) is an important component of the complex gene regulatory network that underlies arabidopsis shoot apical meristem transitions. *Mol. Plant* 8, 796–813. doi: 10.1016/j.molp.2015.01.017

Pot, C., Jin, H., Awasthi, A., Liu, S. M., Lai, C.-Y., Madan, R., et al. (2009). Cutting edge: IL-27 induces the transcription factor c-Maf, cytokine IL-21, and the costimulatory receptor ICOS that coordinately act together to promote differentiation of IL-10-producing Tr1 cells. *J. Immunol.* 183, 797–801. doi: 10.4049/jimmunol.0901233

Puniya, B. L., Todd, R. G., Mohammed, A., Brown, D. M., Barberis, M., and Helikar, T. (2018). A mechanistic computational model reveals that plasticity of CD4+ T cell differentiation is a function of cytokine composition and dosage. *Front. Physiol.* 9:462. doi: 10.3389/fphys.2018.00878

Roncarolo, M. G., Gregori, S., Battaglia, M., Bacchetta, R., Fleischhauer, K., and Levings, M. K. (2006). Interleukin-10-secreting type 1 regulatory T cells in rodents and humans. *Immunol. Rev.* 212, 28–50. doi: 10.1111/j.0105-2896.2006.00420.x

Schmitt, E., Klein, M., and Bopp, T. (2014). Th9 cells, new players in adaptive immunity. *Trends Immunol.* 35, 61–68. doi: 10.1016/j.it.2013.10.004

Sozzani, S., Del Prete, A., and Bosisio, D. (2017). Dendritic cell recruitment and activation in autoimmunity. *J. Autoimmun.* 85, 126–140. doi: 10.1016/j.jaut.2017.07.012

Swain, S. L., Weinberg, A. D., English, M., and Huston, G. (1990). IL-4 directs the development of Th2-like helper effectors. *J. Immunol.* 145, 3796–3806.

Szabo, S. J., Kim, S. T., Costa, G. L., Zhang, X., Fathman, C., and Glimcher, L. H. (2000). A novel transcription factor, T-bet, directs Th1 lineage commitment. *Cell* 100, 655–669. doi: 10.1016/S0092-8674(00)80702-3

Szabo, S. J., Sullivan, B. M., Peng, S. L., and Glimcher, L. H. (2003). Molecular mechanisms regulating Th1 immune responses. *Annu. Rev. Immunol.* 21, 713–758. doi: 10.1146/annurev.immunol.21

Travis, M. A., and Sheppard, D. (2014). TGF-β activation and function in immunity. *Annu. Rev. Immunol.* 32, 51–82. doi: 10.1146/annurev-immunol-032713-120257

Veldhoen, M., Hocking, R. J., Atkins, C. J., Locksley, R. M., and Stockinger, B. (2006). TGFβ in the context of an inflammatory cytokine milieu supports de novo differentiation of IL-17-producing T cells. *Immunity* 24, 179–189. doi: 10.1016/j.immuni.2006.01.001

Verbist, K. C., Guy, C. S., Milasta, S., Liedmann, S., Kaminski, M. M., Wang, R., et al. (2016). Metabolic' maintenance of cell asymmetry following division in activated T lymphocytes. *Nature* 532, 389–393. doi: 10.1038/nature17442

Villarreal, C., Padilla-Longoria, P., and Alvarez-Buylla, E. R. (2012). General theory of genotype to phenotype mapping : derivation of epigenetic landscapes from N-node complex gene regulatory networks. *Phys. Rev. Lett.* 109:118102. doi: 10.1103/PhysRevLett.109.118102

Wei, L., Laurence, A., and O'Shea, J. J. (2008). New insights into the roles of Stat5a/b and Stat3 in T cell development and differentiation. *Semin. Cell Dev. Biol.* 19, 394–400. doi: 10.1016/j.semcdb.2008.07.011

Wittmann, D. M., Krumsiek, J., Saez-Rodriguez, J., Lauffenburger, D. A., Klamt, S., and Theis, F. J. (2009). Transforming Boolean models to continuous models: methodology and application to T-cell receptor signaling. *BMC Syst. Biol.* 3:98. doi: 10.1186/1752-0509-3-98

Wohlfert, E. A., Grainger, J. R., Bouladoux, N., Konkel, J. E., Oldenhove, G., Ribeiro, C. H., et al. (2011). GATA3 controls Foxp3 + regulatory T cell fate during inflammation in mice. *J. Clin. Invest.* 121, 4503–4515. doi: 10.1172/JCI57456

Xu, L., Kitani, A., Fuss, I., and Strober, W. (2007). Cutting edge: regulatory T cells induce CD4 + CD25- Foxp3- T cells or are self-induced to become Th17 cells in the absence of exogenous TGF-. *J. Immunol.* 178, 6725–6729. doi: 10.4049/jimmunol.178.11.6725

Yang, X. O., Nurieva, R., Martinez, G. J., Kang, H. S., Chung, Y., Pappu, B. P., et al. (2008). Molecular antagonism and plasticity of regulatory and inflammatory T cell programs. *Immunity* 29, 44–56. doi: 10.1016/j.immuni.2008.05.007

Yu, D., Rao, S., Tsai, L. M., Lee, S. K., He, Y., Sutcliffe, E. L., et al. (2009). The transcriptional repressor Bcl-6 directs T follicular helper cell lineage commitment. *Immunity* 31, 457–468. doi: 10.1016/j.immuni.2009.07.002

Zadeh, L. A. (1965). Fuzzy sets. *Information Control* 8, 338–353. doi: 10.1016/S0019-9958(65)90241-X

Zheng, W. F., and Flavell, R. A. (1997). The transcription factor GATA-3 is necessary and sufficient for Th2 cytokine gene expression in CD4 T cells. *Cell* 89, 587–596. doi: 10.1016/S0092-8674(00)80240-8

Zheng, Y., Josefowicz, S. Z., Kas, A., Chu, T.-T., Gavin, M. A., and Rudensky, A. Y. (2007). Genome-wide analysis of Foxp3 target genes in developing and mature regulatory T cells. *Nature* 445, 936–940. doi: 10.1038/nature05563

Zhou, L., Ivanov, I. I., Spolski, R., Min, R., Shenderov, K., Egawa, T., et al. (2007). IL-6 programs T H -17 cell differentiation by promoting sequential engagement of the IL-21 and IL-23 pathways. *Nat. Immunol.* 8, 967–974. doi: 10.1038/ni1488

Zhu, J., Yamane, H., and Paul, W. E. (2010). Differentiation of effector CD4 T cell populations (*). *Annu. Rev. Immunol.* 28, 445–489. doi: 10.1146/annurev-immunol-030409-101212

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

TECHNOLOGY REPORT
published: 14 August 2018
doi: 10.3389/fphys.2018.01046

CANA: A Python Package for Quantifying Control and Canalization in Boolean Networks

Rion B. Correia[1,2,3], *Alexander J. Gates*[4], *Xuan Wang*[1] *and Luis M. Rocha*[1,3]*

[1] School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN, United States, [2] CAPES Foundation, Ministry of Education of Brazil, Brasília, Brazil, [3] Instituto Gulbenkian de Ciência, Oeiras, Portugal, [4] Center for Complex Networks Research, Northeastern University, Boston, MA, United States

Logical models offer a simple but powerful means to understand the complex dynamics of biochemical regulation, without the need to estimate kinetic parameters. However, even simple automata components can lead to collective dynamics that are computationally intractable when aggregated into networks. In previous work we demonstrated that automata network models of biochemical regulation are highly canalizing, whereby many variable states and their groupings are redundant (Marques-Pita and Rocha, 2013). The precise charting and measurement of such canalization simplifies these models, making even very large networks amenable to analysis. Moreover, canalization plays an important role in the control, robustness, modularity and criticality of Boolean network dynamics, especially those used to model biochemical regulation (Gates and Rocha, 2016; Gates et al., 2016; Manicka, 2017). Here we describe a new publicly-available Python package that provides the necessary tools to extract, measure, and visualize canalizing redundancy present in Boolean network models. It extracts the pathways most effective in controlling dynamics in these models, including their *effective graph* and *dynamics canalizing map*, as well as other tools to uncover minimum sets of control variables.

Keywords: Boolean networks, automata, canalization, python package, biochemical regulation, logical modeling, network dynamics, complex systems

OPEN ACCESS

Edited by:
Tomáš Helikar,
University of Nebraska-Lincoln, United States

Reviewed by:
Jongrae Kim,
University of Leeds, United Kingdom
Anatoly Sorokin,
Institute of Cell Biophysics (RAS), Russia

***Correspondence:**
Luis M. Rocha
rocha@indiana.edu

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: *06 March 2018*
Accepted: *13 July 2018*
Published: *14 August 2018*

Citation:
Correia RB, Gates AJ, Wang X and Rocha LM (2018) CANA: A Python Package for Quantifying Control and Canalization in Boolean Networks. Front. Physiol. 9:1046. doi: 10.3389/fphys.2018.01046

1. A TOOL TO STUDY REDUNDANCY AND CONTROL IN BOOLEAN NETWORKS

Mathematical and computational modeling of biological networks promises to uncover the fundamental principles of living systems in an integrative manner (Iyengar, 2009; Ideker and Nussinov, 2017). In particular, Boolean Networks (BN), a class of logical dynamical systems, provide an effective framework to capture the dynamics of interconnected biological systems without the need for detailed kinetic parameters (Bornholdt, 2005; Assmann and Albert, 2009). BN have been used to model and predict biochemical regulation in genetic networks (Li et al., 2004), cell signaling (Helikar et al., 2008), chemical reactions in metabolic networks (Chechik et al., 2008), anticancer drug response (Choi et al., 2017), action potentials in neural networks (Kurten, 1988), and many other dynamical systems involved in biomedical complexity (Albert and Othmer, 2003).

Two reasons contribute to the success of BN models: (i) the reduction of complex multivariate dynamics to a graph revealing the organization and constraints of the topology of interactions in biological systems, and (ii) a coarse-grained treatment of dynamics

that facilitates predictions of limiting behavior and robustness (Bornholdt, 2008). However, more than understanding the organization of complex biological systems, we need to derive control strategies that allow us, for example, to intervene on a diseased cell (Zhang et al., 2008), or revert a mature cell to a pluripotent state (Wang and Albert, 2011). Recently, several mathematical tools were developed to enhance our understanding of BN control by removing redundant pathways, identifying key dynamic modules (Marques-Pita and Rocha, 2013), and characterizing critical driver variables (Gates and Rocha, 2016).

Here we present CANA[1], a python package to study redundancy and control in BN models of biochemical dynamics (Correia et al., 2018). It provides a simple interface to access computational tools for three important aspects of BN analysis and prediction:

1. **Dynamics.** Python classes are included to enumerate all *attractors* and calculate the full *state transition graph* (STG) of BN, as described in section 2.
2. **Canalization.** The redundancy properties of automata functions have been characterized as a form of canalization (Kauffman, 1984), particularly when used to model dynamical interactions in models of genetic regulation and biochemical signaling (Kauffman et al., 2004; Reichhardt and Bassler, 2007; Marques-Pita and Rocha, 2013). At the level of individual Boolean transition functions (network nodes), canalization is observed when not all inputs are necessary to determine a state transition (see section 3 for formal definition). CANA can be used to calculate all measures of canalization that derive from removing dynamical redundancy via two-symbol schemata re-description (Marques-Pita and Rocha, 2013): *effective connectivity*, *input redundancy*, and *input symmetry*. At the network level, CANA also calculates the *effective graph*, a weighted and directed graph whose edge weights denote their effective contribution to node transitions, as well as the *dynamics canalizing map*, a parsimonious representation of the necessary and sufficient state transitions that define the entire dynamics of BN. All canalization measures and network representations are applicable to synchronous and asynchronous BN models, as described in section 3.
3. **Control.** From a subset of driver variables—nodes that act as the loci of control interventions—CANA computes the *controlled state transition graph* (CSTG), as well as the *controlled attractor graph* (CAG) capturing all controlled transitions between attractors possible via driver variable interventions (Gates and Rocha, 2016). CANA also computes measures of controllability that depend on the CSTG and CAG: *mean fraction of reachable configurations*, *mean fraction of controlled configurations*, and *mean fraction of reachable attractors*, as described in section 4. Currently, control analysis in CANA is applicable only to synchronous BN models.

Here we demonstrate the full functionality of the CANA package using the BN model of floral organ development in the flowering plant *Arabidopsis thaliana* (Chaos et al., 2006). Additionally, we provide an interface between CANA and the *Cell Collective* (Helikar et al., 2012), allowing for an extensive analysis of control and canalization in complex biological systems.

The CANA package fills a key void in the available library of computational software to analyze Boolean Network models. Existing software falls into two categories: either they are designed to reverse engineer BN models from biological experimental data, or they focus on simulating BN dynamics. Examples of the first category include the *CellNetOptimizer* which creates BN from high-throughput biochemical data (Terfve et al., 2012), and the *Dynamic Deterministic Effects Propagation Networks* (DDEPN) package which reconstructs signaling networks based from time-course experimental data (Bender et al., 2010). The second category of BN simulation packages is best exemplified by *BooleanNet*, a python package that simulates both synchronous and asynchronous dynamics (Albert et al., 2008), and PANET, a Cytoscape plugin that quantifies the robustness of BN models (Trinh et al., 2014). *The Cell Collective*, a collaborative platform and intuitive visual interface to share and build BN models, can also be used to simulate BN dynamics (Helikar et al., 2012). The CANA package expands the set of available tools of the second category, by providing Python classes to calculate measures and visualizations of canalization (dynamical redundancy) and control of BN models, as detailed below. CANA is designed as a toolbox for both computational and experimental system biologists. It enables the simplification of BN models and testing of network control algorithms, thus prioritizing biochemical variables more likely to be relevant for specific biological questions (e.g., genes controlling cell fate), and ideal candidates for knockout experiments.

2. BOOLEAN NETWORK REPRESENTATION AND DYNAMICS

A *Boolean automaton* is a binary variable, $x \in \{0,1\}$, whose state is updated in discrete time-steps, t, according to a deterministic *Boolean state-transition function* of k inputs: $x^{t+1} = f(x_1^t, ..., x_k^t)$. The state-transition function, $f : \{0,1\}^k \rightarrow \{0,1\}$, is defined by a *look-up (truth) table* (LUT), $F \equiv \{f_\alpha : \alpha = 1, ..., 2^k\}$, with one entry for each of the 2^k combinations of input states and a mapping to the automaton's next state (transition or output), x^{t+1} (**Figure 1A**). In CANA, a Boolean automaton—a python class denoted *BooleanNode*—is instantiated from the list of transitions that define its LUT.

A *Boolean Network* is a graph $\mathcal{B} \equiv (X, C)$, where X is a set of N Boolean automata *nodes* $x_i \in X, i = 1, ..., N$ and C is a set of directed edges $c_{ji} \in C : x_i, x_j \in X$ that represent the interaction network, denoting that automaton x_j is an input to automaton x_i, as computed by F_i. The set of inputs for automaton x_i is denoted by $X_i = \{x_j \in X : c_{ji} \in C\}$, and its cardinality, $k_i = |X_i|$, is the *in-degree* of node x_i. At any given time t, $\mathcal{B}$ is in a specific configuration of automata states, $\boldsymbol{x}^t = \langle x_1^t, x_2^t, ..., x_N^t \rangle$, where we use the terms *state* for individual automata (x_i^t) and *configuration* ($\boldsymbol{x}^t$) for the collection of states of all automata of

[1]**CANA**lization: Redundancy & Control in Boolean Networks. For documentation and tutorials (see available online at: github.com/rionbr/CANA)

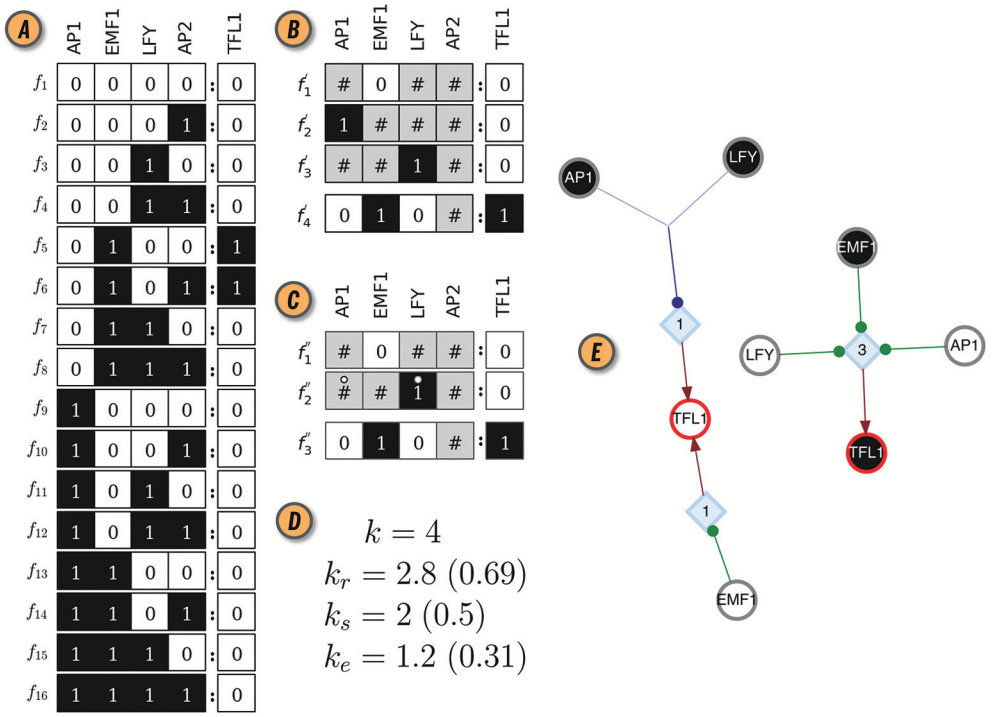


FIGURE 1 | CANA analysis of the Boolean automaton defining the dynamics of the TFL1 gene in the BN model of the floral organ arrangement in the flowering plant *Arabidopsis Thaliana*. **(A)** Look-up-table (LUT). **(B)** Wildcard schema redescription, F'(TFL1). Wildcards are denoted by gray states. As an example, schema f'_4 redescribes the subset of LUT entries $\Upsilon_4 \equiv \{f_5, f_6\}$, where the input variable AP2 can be either *on* or *off*. **(C)** Two-symbol schema redescription, F''(TFL1). Permutation of the inputs marked with the position-free symbol (○) in any schema of F''(TFL1) result in a wildcard schema in F'(TFL1). For example, f''_2 redescribes $\Theta'_2 \equiv \{f'_2, f'_3\}$. **(D)** In-degree ($k$), input redundancy ($k_r$), input symmetry ($k_s$), and effective connectivity (k_e) of TFL1 automaton. Values in parenthesis are the respective (relative) measures normalized by k, used for comparisons between automata with different number of inputs. **(E)** Canalizing Map (CM) of automaton TFL1, with its two possible states, TFL1 $\in \{0, 1\}$, shown as circles with red contour; white (black) fill color denotes state 0 (1). Input variables and their respective state are also shown as circles (*s-units*) with the same color criterion, and link to *t-units* shown as blue diamonds with corresponding threshold value inside; thus, TFL1 requires 3 input conditions (LFY $= 0 \wedge$ EMF1 $= 1 \wedge$ AP1 $= 0$) to turn on (TLF1 $= 1$), but only one (EMF1 $= 0 \vee$ AP1 $= 1 \vee$ LFY $= 1$) to turn off (TLF1 $= 0$); $\wedge$ and $\vee$ denote the logical conjunction (and) and disjunction (or), respectively. Network rendering generated with Graphviz (Ellson et al., 2002).

the BN at time t, i.e. the collective network state. The set of all possible network configurations is denoted by $\mathcal{X} \equiv \{0,1\}^N$, where $|\mathcal{X}| = 2^N$. The dynamics of $\mathcal{B}$ unfolds from an initial configuration, $\boldsymbol{x}^0$, by a *synchronous*, update policy in which all automata transition to the next state at the same time step, or an *asynchronous* update policy, in which automata update their next step in distinct time steps according to some update schedule (e.g. stochastically). The complete dynamical behavior of the system for all initial conditions is captured by the *state-transition graph* (STG), $\mathcal{G} \equiv \text{STG}(\mathcal{B}) = (\mathcal{X}, \mathcal{T})$, where each node is a configuration $\boldsymbol{x}_\alpha \in \mathcal{X}$, and an edge $T_{\alpha,\beta} \in \mathcal{T}$ denotes that a BN in configuration $\boldsymbol{x}_\alpha$ at time t will be in configuration $\boldsymbol{x}_\beta$ at time $t+1$. Under deterministic dynamics, only a single transition edge $T_{\alpha,\beta}$ is allowed out of every configuration node $\boldsymbol{x}_\alpha$. Configurations that repeat, such that $\boldsymbol{x}_\alpha^{t+\mu} = \boldsymbol{x}_\beta^t$, are known as *attractors* and differentiated as *fixed-point* attractors when $\mu = 1$, and *limit cycles* when $\mu > 1$, respectively. Because $\mathcal{G}$ is finite, it contains at least one attractor, as some configuration or limit cycle must repeat in time (Wuensche, 1998).

In CANA, a python class named *BooleanNetwork* represents a BN, and is instantiated from a dictionary containing the transition functions (LUT) of all its constituent automata nodes, or loaded from a file. We also provide several predefined example BN models that can be directly loaded: the *Arabidopsis Thaliana* gene regulatory network (GRN) of flowering patterns (Chaos et al., 2006), a simplified version of the segment polarity GRN of *Drosophila melanogaster* (Albert and Othmer, 2003), the *Budding Yeast* cell-cycle regulatory network (Li et al., 2004), and the BN motifs analyzed in Gates and Rocha (2016). Beyond the aforementioned networks, our current release also incorporates all publicly available networks in the Cell Collective repository (Helikar et al., 2012). These were loaded from the Cell Collective API and converted into truth tables that can be read by CANA[2]. Our package has two built-in methods available to compute network dynamics: for relatively small BN ($N < 30$) the full state-space can be computed, whereas for larger BN, CANA uses a Boolean satisfiability (SAT-based) algorithm, capable of enumerating all attractors in a BN with thousands of variables (Dubrova and Teslenko, 2011).

[2]Future releases will provide a direct link to the Cell Collective API for conversion of Cell Collective models. Currently, models are converted to .CNET (truth table) format, and subsequently imported to CANA.

3. CANALIZATION

Important insights about BN dynamics are gained by observing that not all inputs to an automaton are equally important for determining its state transitions, a concept known as *canalization* (Reichhardt and Bassler, 2007). Originally, the term was proposed by Waddington (1942) and subsequently refined to characterize the buffering of genetic and epigenetic perturbations leading to the stability of phenotypic traits (Siegal and Bergman, 2002; Masel and Maughan, 2007; ten Tusscher and Hogeweg, 2009). Understanding how canalization occurs in a given BN model allows us to uncover and remove redundancy present in the pathways that control its dynamics. In CANA, we follow Marques-Pita and Rocha (2013) by quantifying canalization through the logical *redundancy* present in automata. Specifically, we use the Quine-McCluskey Boolean minimization algorithm (Quine, 1955) to identify those inputs of an automaton which are redundant given the state of its other inputs, thus reducing its LUT to a set of *prime implicants*. The prime implicants are in turn combined to create wildcard schemata, $F' \equiv \{f'_\upsilon\}$, in which the *wildcard* or "Don't care" symbol, # (also represented graphically in gray) denotes an input whose state is redundant given the state of other necessary input states. In this process, the original LUT F (**Figure 1A**) is redescribed by a more compressed set of schemata F' (**Figure 1B**). Every wildcard schema $f'_\upsilon \in F'$ redescribes a subset of entries in the original LUT, denoted by $\Upsilon_\upsilon \equiv \{f_\alpha : f_\alpha \rightarrowtail f'_\upsilon\} \subseteq F$; $\rightarrowtail$ means 'is redescribed by'. Finally, CANA also calculates the *two-symbol schemata* redescription, $F'' \equiv \{f''_\theta\}$, whereby in addition to the wildcard symbol, a *position-free* symbol, ○, further captures *permutation redundancy* (i.e., group-symmetry): subsets of inputs whose states can permute without affecting the automaton's state (**Figure 1C**). Every two-symbol schema $f''_\theta \in F''$ redescribes a set $\Theta_\theta \equiv \{f_\alpha : f_\alpha \rightarrowtail f''_\theta\} \subseteq F$ of LUT entries of automaton x.

Several measures of canalization present in the LUT of an automaton are also defined in CANA, and can be accessed by function calls to both the *BooleanNode* and *BooleanNetwork* classes. *Input redundancy*, $k_r(x)$, measures the number of inputs that on average are not needed to compute the state of automaton x. This is measured by tallying the mean number of wildcard symbols present in the set of schemata $F'(x)$ or $F''(x)$ that redescribe the LUT $F(x)$ (Equation 1). *Effective connectivity*, k_e, is a complementary measure of $k_r(x)$ yielding the number of inputs that are on average necessary to compute the automaton's state (Equation 1). Whereas $k(x)$ is the number of inputs to automaton x present in the BN, $k_e(x)$ is the minimum number of such inputs that are on average necessary to determine the state of x—its effective connectivity or degree. Similarly, *input symmetry*, $k_s(x)$, is the mean number of inputs that can permute without effect on the state of x. It is measured by tallying the mean number of position-free symbols present in $F''(x)$ (Equation 1):

$$k_r(x) = \frac{\sum_{f_\alpha \in F} \max_{\upsilon : f_\alpha \in \Upsilon_\upsilon} \left(n^{\#}_\upsilon\right)}{|F|} \quad , \quad k_e(x) = k(x) - k_r(x) \quad ,$$
$$k_s(x) = \frac{\sum_{f_\alpha \in F} \max_{\theta : f_\alpha \in \Theta_\theta} \left(n^{\circ}_\theta\right)}{|F|} \qquad (1)$$

where $n^{\#}_\upsilon$ and n°_θ are the number of inputs with a # or ○ in schema f'_υ or f''_θ, respectively[3]. **Figure 1D** shows the values of these measures for the LUT of the TFL1 gene in the *thaliana* GRN model. Additional algorithmic details of the two forms of canalization, as well as their importance to study control, robustness, and modularity of BN models of biochemical regulation, are presented in Marques-Pita and Rocha (2013). Next we introduce new per-input measures of canalization as well as the effective graph, which CANA also computes.

Most automata contain redundancy of one or both of the two forms of canalization; only the two parity functions for any k have $k_r = 0$ (e.g., the *XOR* function and its negation for $k = 2$), and even those can have $k_s > 0$. Therefore, the original interaction graph of a BN tends to have much redundancy and does not capture how automata truly influence one another in a BN. To formalize this idea, the CANA package computes an *effective graph*, $\mathcal{E} \equiv (X, E)$, where X is as in section 2 and E is a set of weighted directed edges $e_{ji} \in [0, 1] \forall x_i, x_j \in X$ denoting the *effectiveness* of automaton x_j in determining the truth value of automaton x_i, and computed via Equation 2. Specifically, we define per-input measures of canalization for *redundancy*, *effectiveness*, and *symmetry*, respectively:

$$r_{ji} = \frac{\sum_{f_\alpha \in F_i} \operatorname{avg}_{\upsilon : f_\alpha \in \Upsilon^i_\upsilon} \left(j \rightarrowtail \#\right)_\upsilon}{|F_i|} \quad , \quad e_{ji} = 1 - r_{ji} \quad ,$$
$$s_{ji} = \frac{\sum_{f_\alpha \in F_i} \operatorname{avg}_{\theta : f_\alpha \in \Theta^i_\theta} \left(j \rightarrowtail \circ\right)_\theta}{|F_i|} \qquad (2)$$

where $(j \rightarrowtail \#)_\upsilon$ is a logical condition that assumes the truth value 1(0) if input j is (not) a wildcard in schema f'_υ, and similarly for $(j \rightarrowtail \circ)_\theta$ for a position-free symbol in schema f''_θ; avg is the average operator. Naturally, $k_r(x_i) = \sum_j r_{ji}$, $k_e(x_i) = \sum_j e_{ji}$, and $k_s(x_i) = \sum_j s_{ji}$.

The effective graph was shown to be important in predicting the controllability of BN (Gates and Rocha, 2016). Furthermore, the mean k_e of BN (the mean in-degree of the effective graph) is a better predictor of criticality than the in-degree of the original interaction graph (Manicka, 2017), improving the existing theory for predicting criticality in BN (Aldana, 2003). Those results suggest that Natural Selection can select for canalization, thereby enhancing the stability and controllability of networks with high connectivity, that would otherwise exist in the chaotic regime (Gates et al., 2016; Manicka, 2017). As an example, the interaction and effective graphs of the *Thaliana* GRN BN model, as computed by CANA, are shown in **Figures 2A,B**, demonstrating that much redundancy exists in the original model. The most extreme case of redundancy occurs when an input from x_j to automaton x_i exists in the original interaction graph C, $c_{ji} = 1$, but not in the effective graph $\mathcal{E}$, $e_{ji} = 0$, because

[3] k_r and k_e can be computed on either set of schemata F' (as in Equation 1) or F'' (as in Marques-Pita and Rocha 2013), yielding the same result; k_s must be computed on F''.

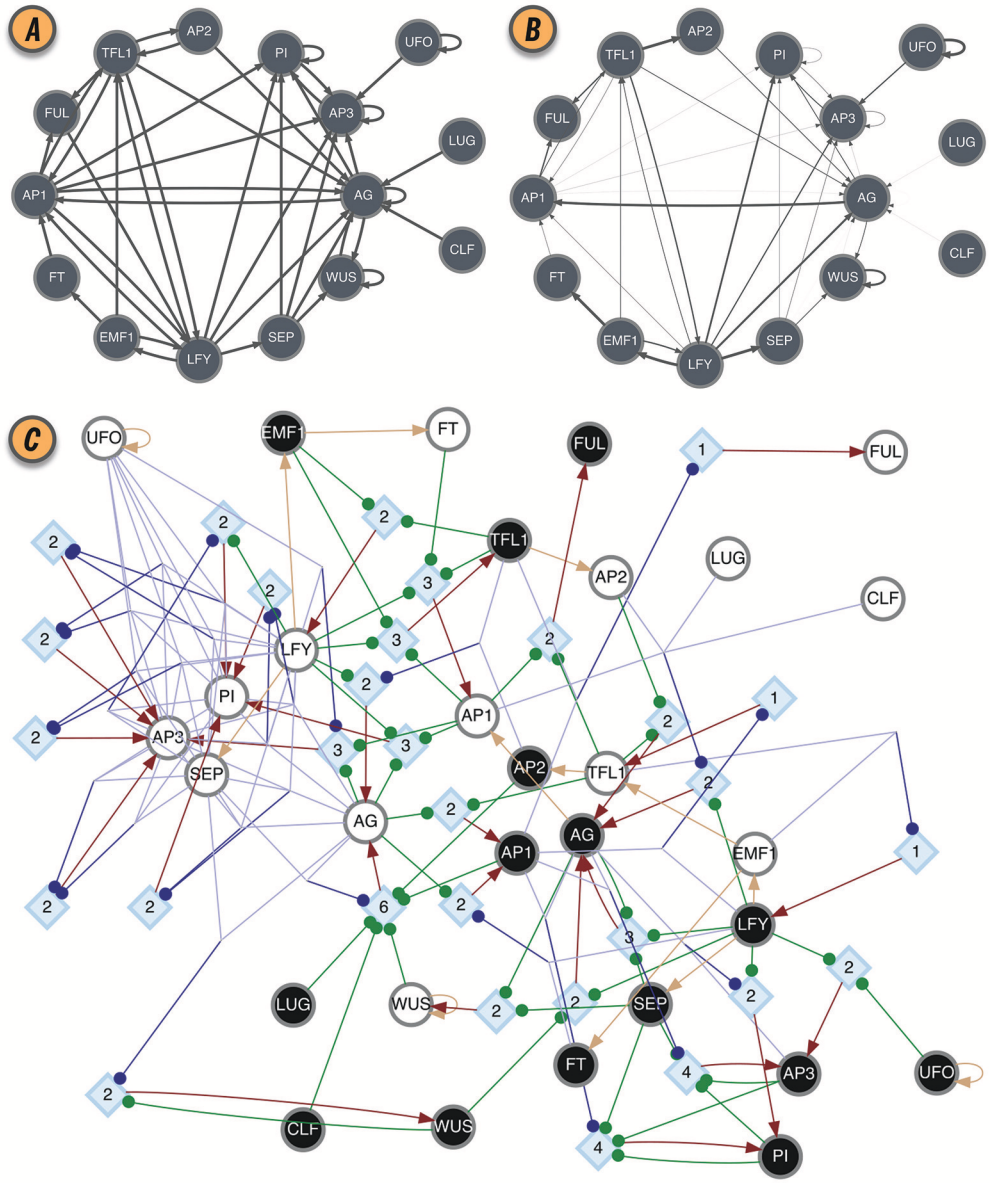


FIGURE 2 | BN model of the floral organ arrangement in the flowering plant *Arabidopsis Thaliana*. **(A)** Interaction graph *C*. **(B)** Effective graph *E*, where edge weights denote e_{ji} (Equation 2). Some edges, originally in *C*, are completely removed in *E* (e.g., AG→AG, AP1→AG, and AP2→TFL1). Others, have very small effectiveness (e.g., AP1→PI and CLF→AG). **(C)** Dynamics Canalization Map (DCM) representing the entire logic of interactions after removal of redundancy. Original BN automata nodes appear twice in the DCM, once for each Boolean truth value and denoted as *s-unit*, white (0) or black (1) circles. When *s-units* are determined by another single *s-unit*, for simplicity and without loss of generality, they are connected with a beige directed edge—a simplification to avoid the rendering of a *t-unit* with a threshold of one. All other variable state determinations occur via *t-units* with larger threshold values. Red edges represent outputs from *t-units* to *s-units*: a state determination of the receiving *s-unit*, after the logical condition of the *t-unit* is met.All other (blue or green) edges denote inputs from *s-units* to *t-units*, that is, the sufficient conditions for a state determination. Blue edges denote group disjunction constraints, whereby conditions captured by *s-units* can merge because any one of the merging conditions is sufficient [e.g., (TFL = 0 ∨ EMF1 = 0) → LFY = 1]. Green edges denote independent and necessary conditions. Directed edges into *s-units* are denoted by arrows, while directed edges into *t-units* are denoted by small circles. Network rendering by Graphviz (Ellson et al., 2002).

it is fully redundant and does not affect the automaton's transition (see several such cases in **Figures 2A,B**).

The canalizing logic of an automaton provided by the schemata set F'', can also be represented as a McCulloch and Pitts (1943) threshold network, named a *Canalizing Map* (CM) in Marques-Pita and Rocha (2013). **Figure 1E** depicts the CM for the TFL1 gene. It consists of two types of nodes: *state units* (*s-unit*, denoted by circles), which represent automata in one of the Boolean truth values ($x_i = 0$, white, or $x_i = 1$, black), and *threshold units* (*t-unit*, denoted by diamonds), which implement a numerical threshold condition on its inputs. When the CM of all automata of a BN are linked, we obtain the

Dynamics Canalization Map (DCM), as shown in **Figure 2C** for the *Thaliana* GRN. Directed fibers connect nodes and propagate an activation pulse; fibers can merge and split, but each end-point always contributes one pulse to an s-unit. The DCM is a highly parsimonious representation of the dynamics of a BN. It contains only necessary information about how (canalizing) control signals determine network dynamics. It enables inferences about control, modularity and robustness to be made about the collective (macro-level) dynamics of BN (Marques-Pita and Rocha, 2013). Because it is assembled using solely the micro-level canalizing logic of individual automata, its computation scales linearly with the number of nodes of the network, and thus it can be computed for very large networks. The computational bottleneck can only be the number of inputs (k) to a particular automaton, since the Quine–McCluskey algorithm grows exponentially with the number of variables. Functions with a large number of variables have to be minimized with heuristic methods such as Espresso (Brayton et al., 1984). Because all measures of canalization, as well as the effective graph and the DCM, derive from removing dynamical redundancy at the level of individual automata, they are independent from the updating regime chosen for the network. In other words, the canalization analysis is applicable to synchronous and asynchronous BN models.

4. CONTROL

The discovery of control strategies in BN models is a central problem in systems biology; theoretical insights about controllability can enhance experimental turnover by focusing experimental interventions on genes and proteins more likely to result in the desired phenotype output. It is well-known that when the set of automata nodes X of a BN is large, enumeration of all configurations $\boldsymbol{x} \in \mathcal{X}$ of its STG becomes difficult, making the controllability of deterministic BN an NP-hard problem (Akutsu et al., 2007). Thus control methodologies which leverage the interaction graph or remove the redundancy in canalizing automata are highly desirable, since they can greatly simplify BN complexity.

CANA contains Python functions designed to provide a testbed for the development of BN control strategies, and to investigate the interplay between canalization, control, and other dynamics properties. Specifically, we study the control exerted on the dynamics of a BN, $\mathcal{B} = (X, C)$, by a subset of *driver variables* $D \subseteq X$—a subset of automata nodes of $\mathcal{B}$. Control *interventions* are realized by instantaneous bit-flip perturbations to the state of the variables in D (Willadsen and Wiles, 2007). This results in a *controlled state transition graph*, CSTG($\mathcal{B}$) $\equiv \mathcal{G}_D \equiv (\mathcal{X}, \mathcal{T} \cup \mathcal{T}_D)$, which is an extension of the STG that captures all possible trajectories due to controlled interventions on D (Gates and Rocha, 2016). The additional edges $\mathcal{T}_D$ denote transitions from every configuration to a set of $2^{|D|} - 1$ configurations in the STG, which are reachable given the bit-flip perturbations of the driver variables. A BN is controllable when every configuration is reachable from every other configuration in $\mathcal{G}_D$ (Sontag, 1998), a condition equivalent to requiring that the CSTG $\mathcal{G}_D$ be strongly connected.

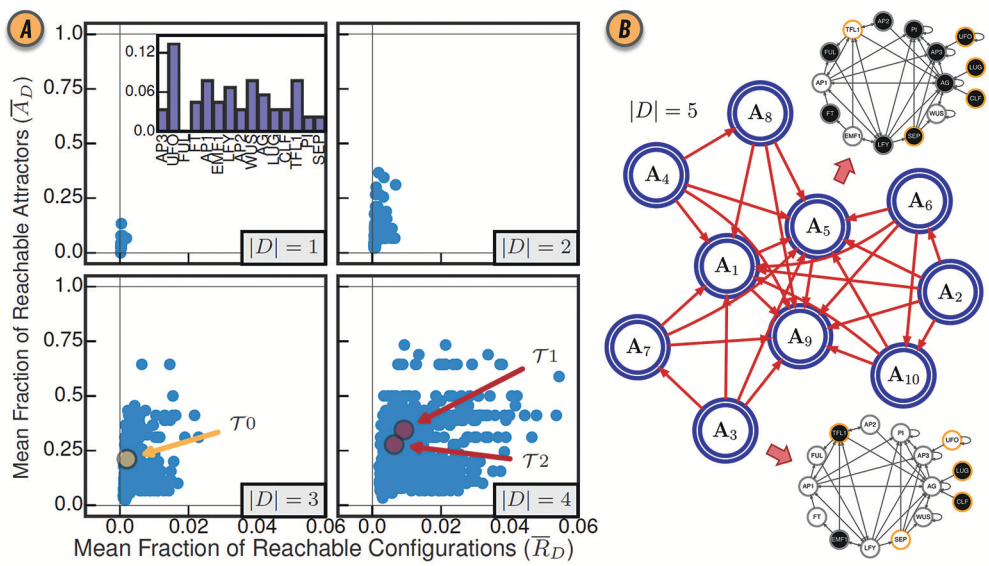


FIGURE 3 | (A) Control of the BN model of gene regulation involved in the floral organ development in the *Arabidopsis thaliana* plant for all driver variable subsets of size $|D| = 1$, $|D| = 2$, $|D| = 3$, and $|D| = 4$. (inset) The mean fraction of reachable attractors $\overline{A}_D$ for each singleton drive variable set. The driver variable subsets predicted by structural controllability to fully control the network are highlighted in red and labeled $\mathcal{T}1$ and $\mathcal{T}2$. The three variable subset with all three root variables is highlighted in yellow and labeled $\mathcal{T}0$. Reproduced from Gates and Rocha (2016) under Creative Commons Attribution 4.0 International (CC BY 4.0) license. **(B)** The CAG with driver variables $D = \{\text{UFO, LUG, CLF, SEP, TFL1}\}$. Each large blue node $A_1, \ldots, A_{10}$ represents an attractor of the network dynamics. The BN configurations for steady-state attractors A_3 and A_5 are shown as interaction graphs with node variables colored white or black for states $x_i = 0$ and $x_i = 1$, respectively; driver variables are shown with a yellow contour.

CANA computes the CSTG of $\mathcal{B}$ given a driver set D, which in turn is used to calculate the *mean fraction of reachable configurations*, $\overline{R}_D$, and the *mean fraction of controlled configurations*, $\overline{C}_D$, (Gates and Rocha, 2016):

$$\overline{R}_D = \frac{1}{2^N} \sum_{\boldsymbol{x}_\alpha \in \mathcal{X}} r(\mathcal{G}_D, \boldsymbol{x}_\alpha) \quad , \quad \overline{C}_D = \overline{R}_D - \overline{R}_\emptyset \quad . \tag{3}$$

where, for each configuration $\boldsymbol{x}_\alpha$, $r(\mathcal{G}_D, \boldsymbol{x}_\alpha)$ is the *fraction of reachable configurations*, defined as the number of other configurations $\mathbf{X}_\beta$ lying on all directed paths from $\boldsymbol{x}_\alpha$, normalized by the total number of other configurations 2^{N-1}. Similarly, the *fraction of controlled configurations* counts the number of new configurations that are reachable due to interventions to D, but were not originally reachable in the STG: $c(\mathcal{G}_D, \boldsymbol{x}_\alpha) = r(\mathcal{G}_D, \boldsymbol{x}_\alpha) - r(\mathcal{G}, \boldsymbol{x}_\alpha)$. When a BN is fully controlled by D, $\overline{R}_D = 1.0$, but for partially controlled BNs $\overline{R}_D \in [0.0, 1.0)$; note that $\overline{C}_D \leq \overline{R}_D$.

In Systems Biology applications, typically only the attractors of BN are meaningful configurations, used to represent different cell types (Kauffman, 1969, 1993; Müller and Schuppert, 2011), diseased or normal conditions (Zhang et al., 2008), and wild-type or mutant phenotypes (Albert and Othmer, 2003). In this context, a more relevant control measure is the extent to which driver variables can steer dynamics from attractor to attractor. To quantify such control, CANA computes the *controlled attractor graph* (CAG) of a BN $\mathcal{B} : \mathcal{C}_D = (\mathcal{A}, \mathcal{Z}_D)$. The nodes of this graph, $\mathbf{A}_\kappa \in \mathcal{A}$, represent an attractor of $\mathcal{B}$, and each edge $z_{\kappa\gamma} \in \mathcal{Z}_D$, denotes the existence of at least one path from attractor $\mathbf{A}_\kappa$ to attractor $\mathbf{A}_\gamma$ in the CSTG $\mathcal{G}_D$ (**Figure 3B**). The *mean fraction of reachable attractors* is then given by

$$\overline{A}_D = \frac{1}{|\mathcal{A}|} \sum_{\mathbf{A}_\kappa \in \mathcal{A}} r(\mathcal{C}_D, \mathbf{A}_\kappa) \tag{4}$$

where $\kappa = 1 \ldots |\mathcal{A}|$ (Gates and Rocha, 2016). Since this notion of control depends only on the enumeration of attractors, CANA can leverage a SAT-based bounded model algorithm to quantify the mean fraction of reachable attractors in a BN with thousands of variables (Dubrova and Teslenko, 2011). **Figure 3A** shows the values of $\overline{R}_D$ and $\overline{A}_D$ for various sizes of driver sets D in the *Thaliana* GRN.

Finally, CANA also provides the functionality to approximate the minimal driver variable subset using two prominent network control methodologies: *Structural Controlability* (SC) (Lin, 1974; Liu et al., 2011) and *Minimum Dominating Set* (MDS) (Nacher, 2012; Nacher and Akutsu, 2013).

5. SUMMARY AND CONCLUSION

We presented a novel, open-source and publicly-available software platform that integrates the analytic methodology used to study canalization in automata network dynamics. This methodology can now be used by others to simplify large automata networks, especially those in models of biochemical regulation dynamics. In addition to the extraction and visualization of specific effective pathways that regulate key phenotypic outcomes in a sea of redundant interaction, CANA includes functionality to measure canalization, uncover control variables, and study dynamical modularity, robustness, and criticality. We hope that the consolidation of redundancy and control algorithms into one package encourages other researchers to build upon our work on canalization, thus adding additional algorithms to CANA.

DATA AVAILABILITY STATEMENT

The CANA python package and all datasets analyzed for this study can be found on Github at github.com/rionbr/CANA.

AUTHOR CONTRIBUTIONS

RC, AG, and XW contributed to the CANA package. LR developed the per-input measures of canalization and the effective graph formulation. RC, AG, and LR wrote the manuscript.

FUNDING

RC was supported by CAPES Foundation grant 18668127, Instituto Gulbenkian de Ciência (IGC), and Indiana University Precision Health to Population Health (P2P) Study. LR was partially funded by the National Institutes of Health, National Library of Medicine Program, grant 01LM011945-01, by a Fulbright Commission fellowship, and by NSF-NRT grant 1735095, Interdisciplinary Training in Complex Networks and Systems. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

ACKNOWLEDGMENTS

We would like to thank Manuel Marques-Pita, Santosh Manicka, and Etienne Nzabarushimana for helpful conversations throughout the development of the CANA package.

REFERENCES

Akutsu, T., Hayashida, M., Ching, W.-K., and Ng, M. K. (2007). Control of Boolean networks: hardness results and algorithms for tree structured networks. *J. Theor. Biol.* 244, 670–679. doi: 10.1016/j.jtbi.2006.09.023

Albert, I., Thakar, J., Li, S., Zhang, R., and Albert, R. (2008). Boolean network simulations for life scientists. *Source Code Biol. Med.* 3:16. doi: 10.1186/1751-0473-3-16

Albert, R., and Othmer, H. G. (2003). The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*. *J. Theor. Biol.* 223, 1–18. doi: 10.1016/S0022-5193(03)00035-3

Aldana, M. (2003). Boolean dynamics of networks with scale-free topology. *Phys. D Nonlinear Phen.* 185, 45–66. doi: 10.1016/S0167-2789(03)00174-X

Assmann, S. M. and Albert, R. (2009). Discrete dynamic modeling with asynchronous update, or how to model complex systems in the

absence of quantitative information. *Methods Mol. Biol* 553, 207–225. doi: 10.1007/978-1-60327-563-7_10

Bender, C., Henjes, F., Fröhlich, H., Wiemann, S., Korf, U., and Beißbarth, T. (2010). Dynamic deterministic effects propagation networks: learning signalling pathways from longitudinal protein array data. *Bioinformatics* 26, i596–i602. doi: 10.1093/bioinformatics/btq385

Bornholdt, S. (2005). Systems biology. Less is more in modeling large genetic networks. *Science* 310, 449–451. doi: 10.1126/science.1119959

Bornholdt, S. (2008). Boolean network models of cellular regulation: prospects and limitations. *J. R. Soc. Interface*, 5(Suppl. 1), S85–S94. doi: 10.1098/rsif.2008.0132.focus

Brayton, R. K., Sangiovanni-Vincentelli, A. L., McMullen, C. T., and Hachtel, G. D. (1984). *Logic Minimization Algorithms for VLSI Synthesis*. Norwell, MA: Kluwer Academic Publishers.

Chaos, A., Aldana, M., Espinosa-Soto, C., de Leon, B. G. P., Arroyo, A. G., and Alvarez-Buylla, E. R. (2006). From genes to flower patterns and evolution: dynamic models of gene regulatory networks. *J. Plant Growth Regul.* 25, 278–289. doi: 10.1007/s00344-006-0068-8

Chechik, G., Oh, E., Rando, O., Weissman, J., Regev, A., and Koller, D. (2008). Activity motifs reveal principles of timing in transcriptional control of the yeast metabolic network. *Nat. Biotechnol.* 26, 1251–1259. doi: 10.1038/nbt.1499

Choi, M., Shi, J., Zhu, Y., Yang, R., and Cho, K.-H. (2017). Network dynamics-based cancer panel stratification for systemic prediction of anticancer drug response. *Nat. Commun.* 8:1940. doi: 10.1038/s41467-017-02160-5

Correia, R. B., Gates, A. J., Wang, X., and Rocha, L. M. (2018). *Canalization: Control & Redundancy in Boolean Networks*. Available online at: https://rionbr.github.io/CANA

Dubrova, E., and Teslenko, M. (2011). A SAT-based algorithm for finding attractors in synchronous boolean networks. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 8, 1393–1399. doi: 10.1109/TCBB.2010.20

Ellson, J., Gansner, E., Koutsofios, L., North, S. C., and Woodhull, G. (2002). "Graphviz—open source graph drawing tools," in *Graph Drawing*, eds P. Mutzel, M. Jünger, and S. Leipert (Berlin; Heidelberg: Springer), 483–484.

Gates, A., Manicka, S., Marques-Pita, M., and Rocha, L. M. (2016). "The effective structure of complex networks drives dynamics, criticality and control," in *Complex Networks 2016: The 5th International Workshop on Complex Networks & Their Applications* (Milan), 107–109.

Gates, A., and Rocha, L. M. (2016). Control of complex networks requires both structure and dynamics. *Sci. Rep.* 6:24456. doi: 10.1038/srep24456

Helikar, T., Konvalina, J., Heidel, J., and Rogers, J. A. (2008). Emergent decision-making in biological signal transduction networks. *Proc. Natl. Acad. Sci. U.S.A.* 105, 1913–1918. doi: 10.1073/pnas.0705088105

Helikar, T., Kowal, B., McClenathan, S., Bruckner, M., Rowley, T., Madrahimov, A., et al. (2012). The cell collective: toward an open and collaborative approach to systems biology. *BMC Syst. Biol.* 6:96. doi: 10.1186/1752-0509-6-96

Ideker, T., and Nussinov, R. (2017). Network approaches and applications in biology. *PLoS Comput. Biol.* 13:e1005771. doi: 10.1371/journal.pcbi.1005771

Iyengar, R. (2009). Why we need quantitative dynamic models. *Sci. Signal.* 2:eg3. doi: 10.1126/scisignal.264eg3

Kauffman, S. (1969). Homeostatis and differentiation in random genetic control networks. *Nature* 224, 177–178.

Kauffman, S. (1993). *The Origins of Order: Self-Organization and Selection in Evolution*. New York, NY: Oxford University Press.

Kauffman, S., Peterson, C., Samuelsson, B., and Troein, C. (2004). Genetic networks with canalyzing boolean rules are always stable. *Proc. Natl. Acad. Sci. U.S.A.* 101, 17102–17107. doi: 10.1073/pnas.0407783101

Kauffman, S. A. (1984). Emergent properties in random complex automata. *Phys. D Nonlinear Phen.* 10, 145–156. doi: 10.1016/0167-2789(84)90257-4

Kurten, K. E. (1988). Correspondence between neural threshold networks and kauffman boolean cellular automata. *J. Phys. A Math. Gen.* 21:L615. doi: 10.1088/0305-4470/21/11/009

Li, F., Long, T., Lu, Y., Ouyang, Q., and Tang, C. (2004). The yeast cell-cycle network is robustly designed. *Proc. Natl. Acad. Sci. U.S.A.* 101, 4781–4786. doi: 10.1073/pnas.0305937101

Lin, C.-T. (1974). Structural controllability. *IEEE Trans. Automatic Control* 19, 201–208. doi: 10.1109/TAC.1974.1100557

Liu, Y.-Y., Slotine, J.-J., and Barabási, A.-L. (2011). Controllability of complex networks. *Nature* 473, 167–173. doi: 10.1038/nature10011

Manicka, S. (2017). *The Role of Canalization in the Spreading of Perturbations in Boolean Networks*. Doctoral dissertation, Indiana University, Informatics and Computing.

Marques-Pita, M., and Rocha, L. M. (2013). Canalization and control in automata networks: body segmentation in drosophila melanogaster. *PLoS ONE* 8:e55946. doi: 10.1371/journal.pone.0055946

Masel, J., and Maughan, H. (2007). Mutations leading to loss of sporulation ability in bacillus subtilis are sufficiently frequent to favor genetic canalization. *Genetics* 175, 453–457. doi: 10.1534/genetics.106.065201

McCulloch, W. S., and Pitts, W. H. (1943). A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophisics* 5, 115–133. doi: 10.1007/BF02478259

Müller, F.-J., and Schuppert, A. (2011). Few inputs can reprogram biological networks. *Nature* 478, E4–E5. doi: 10.1038/nature10543

Nacher, J. C., and Akutsu, T. (2013). Structural controllability of unidirectional bipartite networks. *Sci. Rep.* 3:1647. doi: 10.1038/srep01647

Nacher, J. C., and Akutsu, T. (2012). Dominating scale-free networks with variable scaling exponent: heterogeneous networks are not difficult to control. *N. J. Phys.* 14:073005. doi: 10.1088/1367-2630/14/7/073005

Quine, W. V. (1955). A Way to Simplify Truth Functions. *Am. Math. Monthly* 62, 627–631. doi: 10.1080/00029890.1955.11988710

Reichhardt, C. J. O. and Bassler, K. E. (2007). Canalization and symmetry in boolean models for genetic regulatory networks. *J. Phys. A Math. Theor.* 40:4339. doi: 10.1088/1751-8113/40/16/006

Siegal, M. L., and Bergman, A. (2002). Waddington's canalization revisited: developmental stability and evolution. *Proc. Natl. Acad. Sci. U.S.A.* 99, 10528–10532. doi: 10.1073/pnas.102303999

Sontag, E. D. (1998). *Mathematical Control Theory: Deterministic Finite Dimensional Systems*. New York, NY: Springer.

ten Tusscher, K. H., and Hogeweg, P. (2009). The role of genome and gene regulatory network canalization in the evolution of multi-trait polymorphisms and sympatric speciation. *BMC Evol. Biol.* 9:159. doi: 10.1186/1471-2148-9-159

Terfve, C., Cokelaer, T., Henriques, D., MacNamara, A., Goncalves, E., Morris, M. K., et al. (2012). Cellnoptr: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms. *BMC Syst. Biol.* 6:133. doi: 10.1186/1752-0509-6-133

Trinh, H.-C., Le, D.-H., and Kwon, Y.-K. (2014). Panet: a GPU-based tool for fast parallel analysis of robustness dynamics and feed-forward/feedback loop structures in large-scale biological networks. *PLoS ONE* 9:e103010. doi: 10.1371/journal.pone.0103010

Waddington, C. H. (1942). Canalization of development and the inheritance of acquired characters. *Nature* 150, 563–565.

Wang, R.-S., and Albert, R. (2011). Elementary signaling modes predict the essentiality of signal transduction network components. *BMC Syst. Biol.* 5:44.doi: 10.1186/1752-0509-5-44

Willadsen, K., and Wiles, J. (2007). Robustness and state-space structure of Boolean gene regulatory models. *J. Theor. Biol.* 249, 749–765. doi: 10.1016/j.jtbi.2007.09.004

Wuensche, A. (1998). "Discrete dynamical networks and their attractor basins," in *Complex Systems'98*, eds R. Standish, B. Henry, S. Watt, R. Marks, R. Stocker, D. Green, S. Keen, and T. Bossomaier (Sydney, NSW: University of New South Wales), 1–24.

Zhang, R., Shah, M. V., Yang, J., Nyland, S. B., Liu, X., Yun, J. K., et al. (2008). Network model of survival signaling in large granular lymphocyte leukemia. *Proc. Natl. Acad. Sci. U.S.A.* 105, 16308–16313. doi: 10.1073/pnas.0806447105

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

ORIGINAL RESEARCH
published: 31 August 2018
doi: 10.3389/fphys.2018.01185

Identification of Biologically Essential Nodes via Determinative Power in Logical Models of Cellular Processes

*Trevor Pentzien[1], Bhanwar L. Puniya[2], Tomáš Helikar[2] and Mihaela T. Matache[1]**

[1] Department of Mathematics, University of Nebraska at Omaha, Omaha, NE, United States, [2] Department of Biochemistry, University of Nebraska-Lincoln, Lincoln, NE, United States

A variety of biological networks can be modeled as logical or Boolean networks. However, a simplification of the reality to binary states of the nodes does not ease the difficulty of analyzing the dynamics of large, complex networks, such as signal transduction networks, due to the exponential dependence of the state space on the number of nodes. This paper considers a recently introduced method for finding a fairly small subnetwork, representing a collection of nodes that determine the states of most other nodes with a reasonable level of entropy. The subnetwork contains the most determinative nodes that yield the highest information gain. One of the goals of this paper is to propose an algorithm for finding a suitable subnetwork size. The information gain is quantified by the so-called determinative power of the nodes, which is obtained via the mutual information, a concept originating in information theory. We find the most determinative nodes for 36 network models available in the online database Cell Collective (http://cellcollective.org). We provide statistical information that indicates a weak correlation between the subnetwork size and other variables, such as network size, or maximum and average determinative power of nodes. We observe that the proportion represented by the subnetwork in comparison to the whole network shows a weak tendency to decrease for larger networks. The determinative power of nodes is weakly correlated to the number of outputs of a node, and it appears to be independent of other topological measures such as closeness or betweenness centrality. Once the subnetwork of the most determinative nodes is identified, we generate a biological function analysis of its nodes for some of the 36 networks. The analysis shows that a large fraction of the most determinative nodes are essential and involved in crucial biological functions. The biological pathway analysis of the most determinative nodes shows that they are involved in important disease pathways.

Keywords: Boolean networks, signal transduction network, determinative power, mutual information, simulations, cell collective, gene essentiality, statistical analysis

OPEN ACCESS

Edited by:
Natalia Polouliakh, Sony Computer Science Laboratories, Japan

Reviewed by:
Alessandro Giuliani, Istituto Superiore di Sanità (ISS), Italy
Nicola Bernabò, Università degli Studi di Teramo, Italy
Katsuhiko Murakami, Fujitsu Laboratories (Japan), Japan

***Correspondence:**
Mihaela T. Matache
dmatache@unomaha.edu

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: *19 August 2017*
Accepted: *07 August 2018*
Published: *31 August 2018*

Citation:
Pentzien T, Puniya BL, Helikar T and Matache MT (2018) Identification of Biologically Essential Nodes via Determinative Power in Logical Models of Cellular Processes. Front. Physiol. 9:1185. doi: 10.3389/fphys.2018.01185

1. INTRODUCTION

Boolean networks have gained popularity as models for a variety of real networks where the node activity can be described by two states, 1 and 0, "ON and OFF", "active and non-active," and where each node is updated based on logical relationships with other nodes (e.g., Albert and Thakar, 2014; Abou-Jaoudé et al., 2016). Applications of such models include signal transduction in cells

(e.g., Helikar et al., 2008; Conroy et al., 2014; Abou-Jaoudé et al., 2015; Mendéz and Mendoza, 2016), genetic regulatory networks as well as other biological processes (e.g., Kauffman, 1993; Klemm and Bornholdt, 2000; Shmulevich et al., 2002; Albert and Othmer, 2003; Shmulevich and Kauffman, 2004; Saadatpour et al., 2013).

However, even such a simplification of reality can pose challenges in assessing the dynamics of the network due to the exponential dependence of the state space on the number of nodes. One way to ease the computational burden is to reduce the network to a fairly small subset of nodes that can capture the dynamics of the whole network to a large extent. Some approaches deal with the elimination of nodes that become part of an attractor in the long run, and may also consider removing nodes that are not inputs to any other nodes (Bilke and Sjunnesson, 2001; Richardson, 2004). One can also consider merging or collapsing mediator nodes with one input and one output (Saadatpour et al., 2013). Yet, other approaches consider eliminating irrelevant nodes that are frozen at the same value on every attractor, together with nodes whose outputs go only to irrelevant nodes (Socolar and Kauffman, 2003; Kaufman et al., 2005; Kaufman and Drossel, 2006). In Veliz-Cuba (2011) the author uses a "steady-state approximation" by replacing variables in the Boolean functions governing the nodes' dynamics with their own Boolean expressions, thus reducing the network to a much smaller size that can be used to infer properties about the original network and to gain a better understanding of the role of network topology on the dynamics. In Naldi et al. (2009b) the authors introduce a general method for eliminating nodes sequentially by directly connecting the inputs of a removed node to its output nodes in a manner similar to Veliz-Cuba (2011). Of course, one needs to pay attention and possibly keep nodes that are or may become self-inputs upon elimination of other nodes. The order in which nodes are removed is also important. It is shown that stable states are preserved. In general, attractors may not be preserved. However, the method presented in Saadatpour et al. (2013) is shown to preserve attractors as well.

We consider a recently proposed method for identifying the most powerful nodes in a Boolean network (Heckel et al., 2013; Matache and Matache, 2016). This is done by finding the nodes with the highest determinative power. For a given node, the determinative power is obtained via a summation of all mutual information quantities over all nodes having the given node as a common input. The more powerful the node, the more the information gain provided by the knowledge of its state. The mutual information, as a basic concept in information theory, allows one to represent the reduction of the uncertainty or entropy of the state of a node due to the knowledge of any of its inputs. The entropy has been used in the literature to find the average mutual information of a random Boolean model of regulatory network as a way to quantify the efficiency of information propagation through the entire network (Ribeiro et al., 2008). On the other hand, the entropy of the relevant components of the network, which are comprised of nodes that eventually influence each other's state, has been used as a measure of uncertainty of the future behavior of a random state of the network (Krawitz and Shmulevich, 2007a,b).

In Heckel et al. (2013) it is shown that the knowledge of the states of the most determinative nodes in the feedforward regulatory network of *E. coli* reduces the uncertainty of the overall network significantly. Similar results are observed in Matache and Matache (2016) for a model of general cell signal transduction. It is our goal to explore other models of biological processes obtained from the Cell Collective (http://cellcollective.org), to identify any similarities or differences with respect to previous observations, and to possibly identify any correlations with other network variables or trends in the observed network data. At the same time, we show that the majority of nodes with the most determinative power are essential. Cell Collective provides a variety of gene networks. Essential genes are those genes of an organism that are thought to be critical for its survival and are involved in crucial biological functions.

In section 2, we provide the basic mathematical framework and definitions. We present the algorithm for finding a suitable subnetwork size in section 3. In section 4 we describe the networks under consideration and we provide the results of our simulations paired with a statistical analysis of the data. Then we focus on the analysis of the biological relevance of the most determinative nodes. We provide a discussion of the results in section 5. Conclusions and further directions of research are in section 6.

2. DETERMINATIVE POWER

In this section, we provide the main concepts leading to the determinative power of nodes in a Boolean network.

DEFINITION 1. *Let $\Omega^n = \{0,1\}^n$. A Boolean network (BN) is modelled as a set $[n] := \{1,2,\ldots,n\}$ of n nodes, each node being ON (in state 1) or OFF (in state 0). Then any $\omega \in \Omega^n$ is a possible state of the network. Each node $i \in [n]$ has an associated Boolean function $f_i : \Omega^n \to \Omega$ that governs the dynamics of the node.*

We are usually interested in how the network evolves by iterating the map $F = (f_1, f_2, \ldots, f_n)$ a large number of times.

In this paper, a subnetwork refers to a subset of nodes of the network. One recent approach for finding subnetworks whose nodes determine the states of most other nodes with a reasonable level of entropy focused on the nodes with the most determinative power (DP) (Heckel et al., 2013; Matache and Matache, 2016). The DP is obtained via concepts from information theory. We recall the main definitions and concepts from Cover and Thomas (2006) and Heckel et al. (2013). These include the notion of entropy of random variables, which is a measure of uncertainty, and the mutual information, which is a measure of dependence between two random variables and is defined in terms of the entropy.

DEFINITION 2. *Let X and Y be discrete random variables. The (Shannon) entropy of X is defined as*

$$H(X) = -\sum_x p_x \log_2 p_x = -E[\log_2 P(X)]$$

where x are the values of the random variable X, $p_x = P(X = x)$, and $E[\log_2 P(X)]$ is the expected value of the random variable

$\log_2 P(X)$. *In binary this reduces to the function*

$$h(p) = -p\log_2(p) - (1-p)\log_2(1-p), \quad p = P(X=1), \quad h(0) = h(1) = 0.$$

The conditional entropy of Y conditional on the knowledge of X is

$$H(Y|X) = -E[\log_2 P(Y|X)].$$

The mutual information (MI) is the reduction of uncertainty of the random variable Y due to the knowledge of X. That is

$$MI(Y;X) = H(Y) - H(Y|X).$$

In principle, the mutual information is a measure of the "gain of information," or the determinative power (DP) of X over Y. The authors of Heckel et al. (2013) use the MI to construct the DP of a node j over the states of a Boolean network, namely

$$DP(j) = \sum_{i=1}^{n} MI(f_i(X); X_j) \tag{1}$$

which represents a summation of all "information gains" obtained from node j over its outputs (i.e., nodes i that have j as an input). Here, the states of the nodes are labeled $X_1, X_2, \ldots, X_n$, and $X = (X_1, X_2, \ldots, X_n)$ represents the state of the network. The notation $f_i(X)$ represents the random variable that describes the dynamical rule of node i. Not all variables $X_1, X_2, \ldots, X_n$ are relevant for the computation of $f_i(X)$ since the actual number of inputs may differ from one node to another. The authors identify the nodes with the largest determinative power in a feedforward *E. coli* network, with the goal of finding a subnetwork whose knowledge can provide sufficient information about the entire network; in other words the entropy of the network conditional on the knowledge of that subnetwork is small enough. They show that in the *E. coli* network, one could consider a subnetwork consisting of less than half of the nodes, and that for larger subnetworks, the entropy does not improve significantly once an approximate (threshold) subnetwork size is reached. Similar results have been found in Matache and Matache (2016) for a signal transduction model in fibroblast cells, paired with a mathematical generalization of some of the results in Heckel et al. (2013) under more relaxed assumptions. Our goal is to use a similar approach for other networks to identify if this type of behavior is typical or not. In the next section, we describe the networks under consideration and then we present the algorithm for finding a suitable subnetwork size. However, before we do that, let us provide an example illustrating the computation of DP according to formula (1). The mutual information terms in (1) are obtained using a formula derived in Matache and Matache (2016). We combine Theorem 1 and Proposition 4 of Matache and Matache (2016) in a suitable way to provide a brief explanation of how the formula is obtained.

The mutual information formula $MI(f_i(X); X_j)$ can be written as

$$\begin{aligned} &MI(f_i(X); X_j) \\ &= h\left(\sum_{x\in \operatorname{supp} f_i} p_x\right) - P(X_j=1)h\left(\sum_{x\in \operatorname{supp} f_i} P(X=x|X_j=1)\right) \\ &\quad -P(X_j=0)h\left(\sum_{x\in \operatorname{supp} f_i} P(X=x|X_j=0)\right) \end{aligned} \tag{2}$$

where $\operatorname{supp} f_i = \{x : f_i(x) = 1\}$ is the support of the function f_i, and $P(X = x|X_j = x_j)$ is the conditional probability of $X = x$ given $X_j = x_j$.

The formula follows directly from the definition of the mutual information

$$MI(f_i(X); X_j) = H(f_i(X)) - H(f_i(X)|X_j). \tag{3}$$

Observe that

$$\begin{aligned} H(f_i(X)) &= h(P(f_i(X)=1)) \\ &= h(E[f_i(X)]) \\ &= h\left(\sum_{x\in\{0,1\}^n} f_i(x)p_x\right) = h\left(\sum_{x\in \operatorname{supp} f_i} p_x\right) \end{aligned} \tag{4}$$

where we use the known fact that for a (Bernoulli) random variable B with values 0 and 1, we have that $P(B = 1) = E[B]$. Similarly,

$$\begin{aligned} H(f_i(X)|X_j) &= \sum_{x_j\in\{0,1\}} P(X_j=x_j)H(f_i(X)|X_j=x_j) \\ &= \sum_{x_j\in\{0,1\}} P(X_j=x_j)h\left(P(f_i(X)=1|X_j=x_j)\right). \end{aligned}$$

On the other hand,

$$\begin{aligned} P(f_i(X)=1|X_j=x_j) &= E[f_i(X)|X_j=x_j] \\ &= \sum_{x\in\{0,1\}^n} f_i(x)P(X=x|X_j=x_j) \\ &= \sum_{x\in \operatorname{supp} f_i} P(X=x|X_j=x_j). \end{aligned}$$

This implies

$$\begin{aligned} H(f_i(X)|X_j) &= \sum_{x_j\in\{0,1\}} P(X_j=x_j)h\left(\sum_{x\in \operatorname{supp} f_i} P(X=x|X_j=x_j)\right) \\ &= P(X_j=1)h\left(\sum_{x\in \operatorname{supp} f_i} P(X=x|X_j=1)\right) \\ &\quad + P(X_j=0)h\left(\sum_{x\in \operatorname{supp} f_i} P(X=x|X_j=0)\right). \end{aligned} \tag{5}$$

Replacing formulas (4) and (5) in (3) we obtain formula (2) which we use in the next example.

EXAMPLE 1. *Consider the 4-node network with states* $X = (X_1, X_2, X_3, X_4)$. *For simplicity we assume that* X *is a uniform random variable that assigns equal probabilities to all* x. *Therefore,* $P(X_i = 1) = P(X_i = 0) = 1/2$ *for* $i = 1, 2, 3, 4$. *Define the Boolean rules as follows:*

$$f_1(x_2, x_3, x_4) = x_2 \wedge x_3 \wedge (1 - x_4);$$
$$f_2(x_1, x_2, x_3) = x_1 \wedge (x_2 \vee x_3); \quad f_3(x_1, x_2) = x_1 \vee x_2.$$

Observe that the actual inputs differ from one node to the other, and that X_4 *can be regarded as an external input with one single output* X_1, *and does not have a Boolean update rule* f_4. *We can see that*

$$\operatorname{supp} f_1 = \{(1,1,0)\}; \quad \operatorname{supp} f_2 = \{(1,0,1),(1,1,0),(1,1,1)\};$$
$$\operatorname{supp} f_3 = \{(0,1),(1,0),(1,1)\}.$$

We obtain the following.

Formula (1)	$DP(i)$
$DP(1) = MI(f_2(X); X_1) + MI(f_3(X); X_1)$	$DP(1) = 0.8601$
$DP(2) = MI(f_1(X); X_2) + MI(f_2(X); X_2) + MI(f_3(X); X_2)$	$DP(2) = 0.6714$
$DP(3) = MI(f_1(X); X_3) + MI(f_2(X); X_3)$	$DP(3) = 0.3601$
$DP(4) = MI(f_1(X); X_4)$	$DP(4) = 0.1379$

For example, to find $MI(f_2(X); X_1)$, *we note that* $\sum_{x \in \operatorname{supp} f_2} p_x = 3/8$. *Since all elements of* $\operatorname{supp} f_2$ *have* $X_1 = 1$, *it follows that*

$$\sum_{x \in \operatorname{supp} f_2} P(X = x | X_1 = 0) = 0$$

and

$$\sum_{x \in \operatorname{supp} f_2} P(X = x | X_1 = 1) = \sum_{x \in \operatorname{supp} f_2} \frac{P(X = x, X_1 = 1)}{P(X_1 = 1)}$$
$$= \frac{P(1,0,1)}{1/2} + \frac{P(1,1,0)}{1/2} + \frac{P(1,1,1)}{1/2}$$
$$= \frac{1/8}{1/2} + \frac{1/8}{1/2} + \frac{1/8}{1/2} = \frac{3/8}{1/2} = 3/4$$

due to the assumption of a uniform distribution of the inputs. Then $MI(f_2(X); X_1) = h(3/8) - \frac{1}{2}h(3/4) = 0.5488$. *Similarly,* $MI(f_3(X); X_1) = h(3/4) - \frac{1}{2}(h(1) + h(1/2)) = 0.3113$. *Thus,* $DP(1) = 0.8601$ *and the other DP values are obtained the same way and are included in the last column of the table above. Thus, node 1 is the most determinative in this network, followed by nodes 2, 3, and 4 in that order. This example points out that nodes with most outputs need not be the most determinative due to the Boolean function governing the node dynamics. At the same time, nodes that have the same number of outputs can lead to very different DP values.*

In the numerical results to be presented in this paper, we use the assumption of ergodicity, meaning that all input states are equally likely. Although this may not be a perfect reflection of reality, it is a most common approach in studying the dynamics of Boolean models for biological networks. For example, this assumption is used in Heckel et al. (2013), the paper that introduces the DP concept for identifying the most powerful nodes in a Boolean network. In Heckel et al. (2013) it is shown that the knowledge of the states of the most determinative nodes in the feedforward regulatory network of *E. coli* reduces the uncertainty of the overall network significantly. However, further study of non-ergodic scenarios may provide new insights.

3. SUBNETWORK SIZE

Let us briefly describe the types of networks that will be used in simulations and for which statistical data are collected and analyzed.

The networks are obtained from Cell Collective (CC, www.cellcollective.org, Helikar et al., 2012, 2013), an interactive platform for building and simulating logical models. The database contains over 60 peer-reviewed published models of biological networks and processes. The networks are of many sizes and represent a variety of different biological processing across a number of different organisms [e.g., yeast (Irons, 2009; Todd and Helikar, 2012), flies (Marques-Pita and Rocha, 2013), humans (Conroy et al., 2014; Mendéz and Mendoza, 2016)]. Models can be simulated and analyzed directly in Cell Collective, or downloaded (as SBML or truth table files) for additional analyses in other tools. In our simulations, truth tables for a collection of networks from Cell Collective are formatted and used in a Matlab program to find the DP and subnetwork size using the above equations.

Next, we provide the actual algorithm used in conjunction with the DP of nodes to find a suitable size for the subnetwork consisting of the most determinative nodes.

Once each $DP(j)$ is computed for $j = 1, 2, \ldots, n$, we can sort them to identify the nodes with highest DP values. We provide an example in **Figure 1** (top) where we show the DP values in ascending order for a *T-cell Receptor Signaling* network (Saez-Rodriguez et al., 2007, https://cellcollective.org/#2171/t-cell-receptor-signaling) with 94 nodes (blue curve). We also plot the maximum possible DP values (with dotted red line) given by the total number of outputs of each node, to have an understanding of how the DP compares to this maximum. Observe that if all mutual information terms would take on their maximum possible value of 1, then the DP would be the number of outputs of the node under consideration. By plotting both the DP values and the maximum possible, we can assess the "efficiency" of the node in generating the information gain in the network.

Once the DP values are sorted, we can compute the overall network entropy generated by subnetworks chosen based on top DP values of nodes. For large networks this can become a difficult task. Therefore, following the work of Heckel et al. (2013), we simplify the computations by considering an upper bound for the entropy. If we consider the collection S_l of the top l most

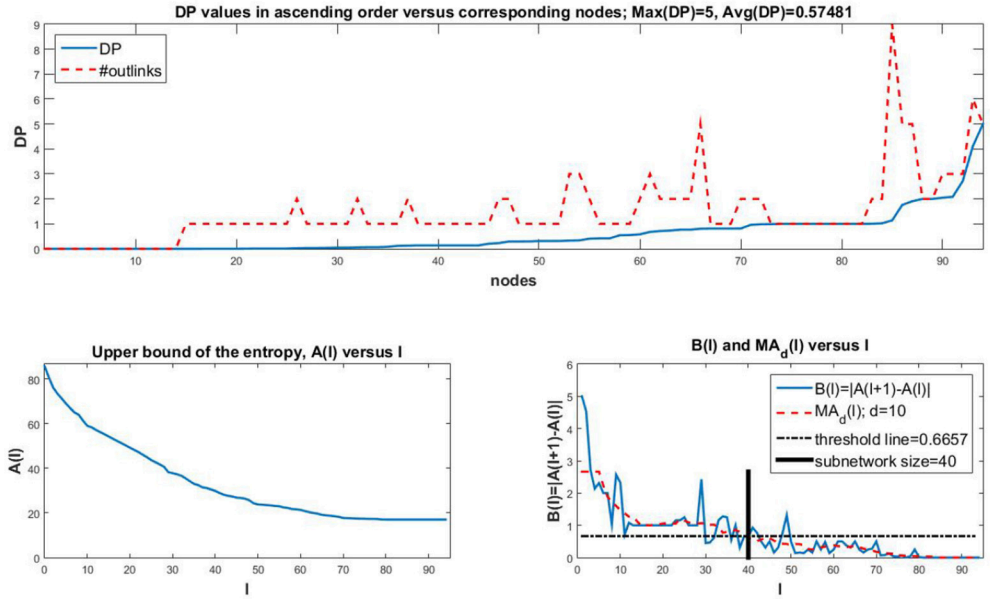


FIGURE 1 | (Top) Sorted DP values (blue curve) for the nodes of a *T-cell Receptor Signaling* of size $n = 94$. The maximum possible values for DP, that is, the number of outputs of each node, are plotted with a red dotted curve for comparison. Notice that most nodes do not reach their maximum possible DP, and that the average DP of 0.57481 is small in comparison to the maximum DP value of 5. Thus, most nodes have a rather small DP. **(Bottom Left)** $A(l)$ vs. l. Observe that the curve has an initial drop for the nodes with the largest DP values, after which the rate of convergence to a positive value is reduced as l increases. This indicates that once a threshold value of l is reached, the entropy does not decrease significantly anymore. Also, the value of $A(l)$ for $l = 1$ is more than 4 times larger than the final value for $l = 94$. **(Bottom Right)** $B(l)$ and $MA_d(l)$ vs. l. We also plot a horizontal line indicating the chosen threshold T for the size of the subnetwork. The bold vertical line segment indicates the subnetwork size. For this particular network, the subnetwork size is less than half of the network size.

determinative nodes, then we can compute

$$H(X|X_{S_l}) \leq \sum_{i=1}^{n} H(X_i|X_{S_l}), \quad \text{for } l = 1, 2, 3, \ldots, n \tag{6}$$

where X_{S_l} is the random variable whose values are the states of the nodes in S_l. In **Figure 1** (bottom left), we plot the values of the larger quantity in (6), namely $A(l) = \sum_{i=1}^{n} H(X_i|X_{S_l})$ which is an upper bound for the entropy of the network given the top l nodes. Observe that for this case, subnetworks of sizes 40–50 or more (with approximation) do not yield a significant improvement of the entropy. Thus it suffices to consider less than half of the original network to be able to predict the overall network behavior with fairly low uncertainty/entropy levels. Observe also that the entropy converges to a positive value as the subnetwork size approaches the network size. This is due to the inherent uncertainty in the network based on its topology and dynamical rules.

In order to identify a precise cutoff for the subnetwork size, we follow the algorithm described next. This algorithm identifies the cutoff observed in **Figure 1** (bottom right; thick vertical line segment).

(I) Start with the sequence $\{A(l), l = 1, 2, \ldots, n\}$.

(II) Construct the associated sequence of distances between consecutive terms of this sequence. That is, construct the sequence $\{B(l) = |A(l+1) - A(l)|, l = 1, 2, \ldots, n-1\}$.

(III) Smooth out the sequence by applying a moving average procedure of order d, which, in our simulations it is set to $0.1(n-1)$ *(rounded up). That is, we consider the averages over d consecutive terms of the sequence. Namely, for* $u = 1, 2, \ldots, (n-1)-(d-1)$, *in other words for* $u = 1, 2, \ldots, n-d$, *the moving average is given by*

$$\frac{1}{d} \sum_{j=l}^{u+d-1} B(j). \tag{7}$$

The first and last elements of the sequence are repeated as necessary so that the final sequence of moving averages has the same length at the original sequence to be averaged. For a given d we label the sequence of moving averages $MA_d = \{MA_d(l), l = 1, 2, \ldots, n-1\}$ *including all terms of formula (7) with the necessary repetitions of the first and last elements to obtain a total of* $n-1$ *terms. An even d value generates an odd number of repeated elements, which leads to one extra repetition of the last element as opposed to the repetitions of the first element (see* MA_4 *in the example below).*

For instance, if the input sequence of $B(l)$ *values is* $\{10, 9, 8, 7, 6, 5, 4, 3, 2, 1\}$ *then some sample* $\{MA_d\}$ *sequences are*

$$MA_3 = \{9, 9, 8, 7, 6, 5, 4, 3, 2, 2\}$$
$$MA_4 = \{8.5, 8.5, 7.5, 6.5, 5.5, 4.5, 3.5, 2.5, 2.5, 2.5\}$$
$$MA_5 = \{8, 8, 8, 7, 6, 5, 4, 3, 3, 3\}.$$

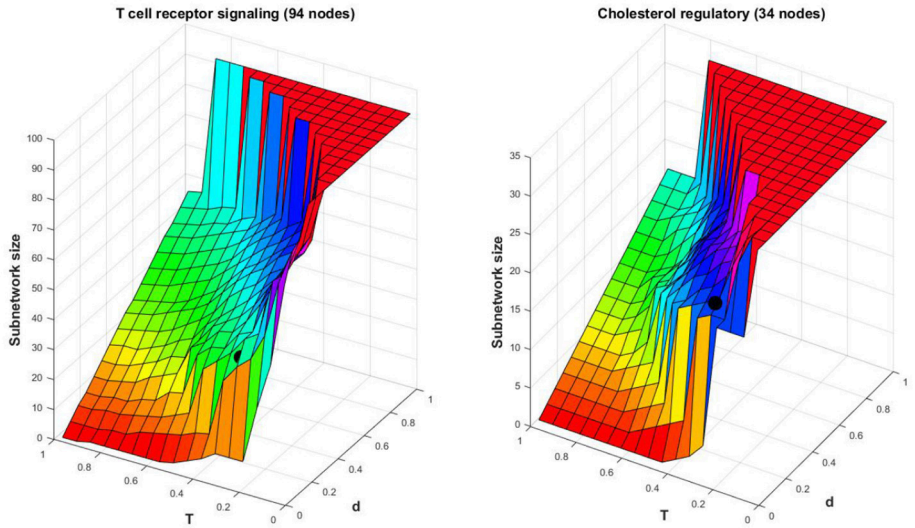


FIGURE 2 | Surface plot for L vs. a grid of values of d and T. The black dot represents the point $L, d = 0.1(n-1), T = \frac{1}{4}\max(MA_d)$ for the *T-cell Receptor Signaling* model with $L = 40$, and for the *Cholesterol Regulatory Pathway* model with $L = 22$.

For example, to clarify even further, in the case of MA_3 and $u = 1$, formula (3) generates $1/3(B(1) + B(2) + B(3)) = 9$. However, since $n - d = 10 - 2 = 8$ we repeat the first and last terms of the sequence given by (3), so that $MA_3(1) = MA_3(2) = 9$. Similarly, $MA_3(9) = MA_3(10) = 1/3(B(8) + B(9) + B(10)) = 2$.

(IV) Set T, the threshold for finding the size of the subnetwork. In simulations we use $T = \frac{1}{4}\max(MA_d)$. More precisely, starting with $l = 1$, we increase l by one unit until we reach a value L for which the following conditions are satisfied

$$MA_d(L) \leq T \quad \text{and} \quad \frac{1}{d}\sum_{j=L}^{\min(L+d-1,n-1)} MA_d(j) \leq T. \quad (8)$$

That is, the values of the MA_d sequence drop below the threshold T and the average variance of the next d values of MA_d is also less than the threshold T.

(V) The subnetwork consists of the nodes with the L highest DP values.

The results are dependent on how one sets the parameters d and T. The larger the d value, the smoother the MA_d sequence, and thus the conditions (8) tend to be satisfied for smaller values of l. The same happens if T is sufficiently large. On the other hand, larger moving average order d means losing some of the intrinsic variation of data. Therefore, we need to be aware of the tradeoff between accuracy and details of the data, as is customary in network modeling and simulation.

In **Figure 1** (bottom right), this algorithm with $d = 0.1(n-1)$ and $T = \frac{1}{4}\max(MA_d)$ generates a minimal subnetwork size of 40 nodes with the largest DP. This is less than half of the network size. We notice that the threshold T is approximately $\frac{1}{4}\max(MA_d) = \frac{1}{4} \cdot 2.8 = 0.7$.

To see how the two parameters d and T affect the size L of the subnetwork, we compute L for a grid of values of d and T for two networks that will be used as examples in the next section too. Two sample surfaces are shown in **Figure 2**. The black dot indicates the actual L value obtained with this procedure for $d = 0.1(n-1)$ and $T = \frac{1}{4}\max(MA_d)$ considered in the simulations. As expected, the values of L increase with an increase of the two parameters, and the surfaces are similar in shape with mild variations.The choice of $d = 0.1(n-1)$ used in simulations generates subnetworks that do not surpass 60% of the network size with approximation. We will see that this is sufficient to identify a good fraction of biologically important nodes in several networks from the Cell Collective (Helikar et al., 2012, 2013).

We explore other networks in the next section, however, we will provide graphs related to the networks considered so far and add one more network of small size.

4. NUMERICAL RESULTS AND ANALYSIS

4.1. Simulations and Statistical Analysis

We apply the procedure explained in the previous section to a number of networks available in Cell Collective (Helikar et al., 2012, 2013). We summarize the results below and supplement with suitably chosen graphs. For each network shown in graphs, we plot the sorted DP values for all nodes, the upper bound for the entropy, $A(l)$ vs. l, and the elements of the algorithm for finding the subnetwork size, namely $B(l)$ and $MA_d(l)$ vs. l with a horizontal line at the threshold value T that indicates the subnetwork size.

The graphs of $A(l)$ consist of a curve that decreases to zero or a value that stabilizes for large values of l in most cases. A typical example is the one considered in the previous section for

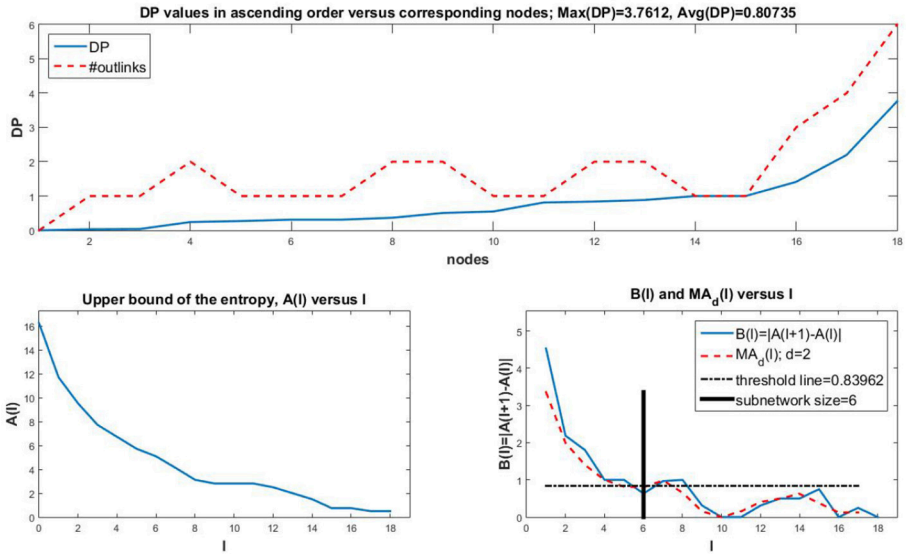


FIGURE 3 | Analog of **Figure 1** for the *Oxidative Stress Pathway* network with $n = 18$. The average DP is larger than for the *T-cell Receptor Signaling* network, which can be expected in a smaller network where nodes may incorporate more information to be used in the network. The maximum DP is smaller though. Observe that here, $A(l)$ decreases to a value close to zero along a non-linear curve. The subnetwork size is a third of the network size, so it is smaller as a fraction of the network in comparison to the *T-cell Receptor Signaling* network where the subnetwork is about 42% of the network size.

the *T-cell Receptor Signaling* network in **Figure 1**. This behavior is very similar to the results obtained in Heckel et al. (2013), the paper that inspired this work, for a feedforward regulatory network in *E. coli*. Notice that $A(l)$ stabilizes at a positive value for large l and does not converge to zero. In general, since $A(l)$ is an upper bound for the entropy as seen in inequality (6), it may not approach zero. On the other hand, the entropy itself is the expected value of a random variable as indicated in Definition 2, and therefore it may be non-zero.

A couple of variations are shown as well. In **Figure 3** we consider a small *Oxidative Stress Pathway* network with 18 nodes (Sridharan et al., 2012, https://cellcollective.org/#3512/oxidative-stress-pathway). The subnetwork size is a third of the network size. In **Figure 4** we show similar graphs for a medium sized *Cholesterol Regulatory Pathway* network with 34 nodes (Kervizic and Corcos, 2008, https://cellcollective.org/#2172/cholesterol-regulatory-pathway). In this case, the upper bound $A(l)$ approaches zero rather slowly at an almost linear rate, therefore the subnetwork size is larger when compared to the whole network, namely about 65% of the entire network.

Next, we summarize the data obtained from a total of 36 networks and generate some statistical information. Four networks are significantly larger than the others: signal transduction in fibroblast cells with 130 nodes, interleukin-1 signaling with 103 nodes, signal transduction in a macrophage with 302 nodes, and T-cell receptor signaling with 94 nodes. We consider them "outliers" and explore some statistics on the remaining 32 networks to avoid skewed results. We hope to be able to expand the list of large networks in the future and include them in the analysis.

We provide boxplots for seven numerical characteristics obtained from the network data: network size n, subnetwork size L, maximum DP values, average DP values, ratio L/n, number of links or edges in the network, E, given by the total number of inputs or outputs for all nodes, and E/n^2 as the ratio between the edges and total number of possible edges, taking into account that self-inputs are allowed. The results are shown in **Figure 5**. We choose to separate them due to the different ranges of values. Observe that most subnetwork sizes are fairly small even for larger networks or more edges, so the subnetwork sizes may not increase with the network size or the number of edges. The number of nodes and the number of edges have similar boxplots. The maximum DP can be fairly large; however it is not clear yet if this fact is related to the network size, or the number of edges. We will explore the idea in what follows. Finally, the average DP is rather small for all networks, regardless of their sizes. Also, most of the ratios L/n of the subnetwork size vs. the network size are less than 60%.

We also explore the dependencies between the numerical characteristics considered in **Figure 5**, by generating a number of scatter plots with corresponding fitted regression lines. In particular, we want to see if there are correlations between $L, L/n$ or the maximum DP and average DP vs. the network parameters $n, E, E/n^2$. We find that there is no evidence of strong correlations between the variables, except for L vs. n, E and maximum DP vs. n, E. The scatter plots are shown in **Figure 6** and the corresponding fitted lines and coefficients of determination R^2 are listed in **Table 1**. Note that there is no strong linear (or non-linear) relationship; however we note the increasing trend in both subnetwork size L and maximum DP with increased n and E. On the other hand we see that the average DP does not depend on the

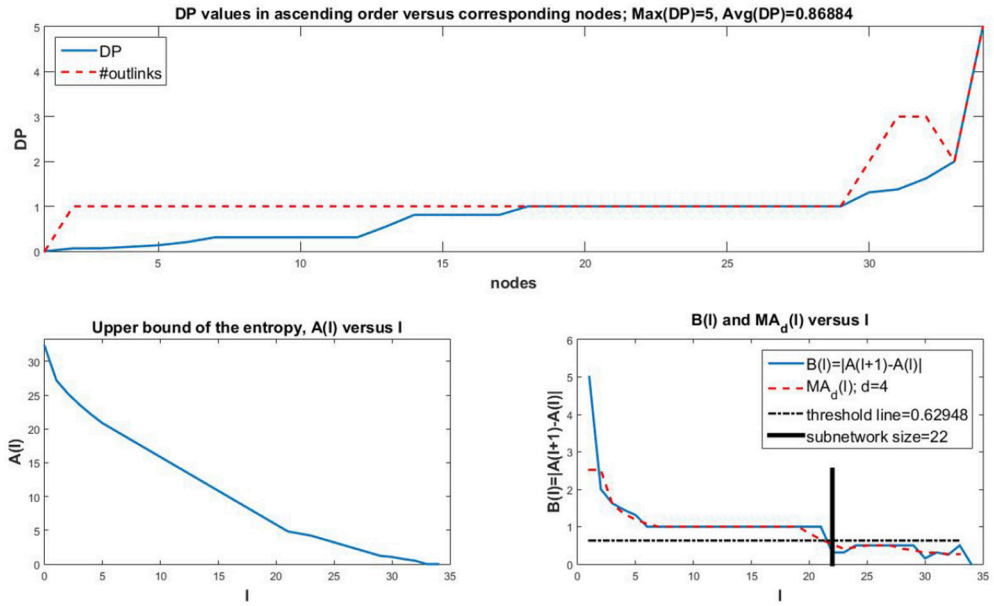


FIGURE 4 | The *Cholesterol Regulatory Pathway* network consists of $n = 34$ nodes. The maximum and average DP are comparable to the *Oxidative Stress Pathway* network. The upper bound $A(l)$ for the *Cholesterol Regulatory Pathway* network has an almost linear decrease to zero. Therefore, the subnetwork size of 22 is larger than in previous cases in comparison to the network size, representing about 65% of the network.

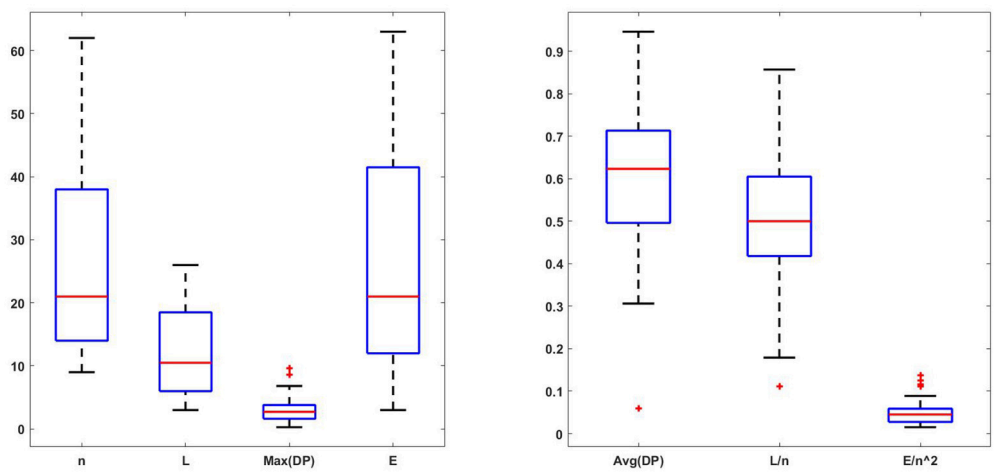


FIGURE 5 | Boxplots for the network size n, subnetwork size L, maximum DP, average DP, ratio L/n, number of edges E, and the ratio E/n^2. They are grouped based on similar magnitudes. The boxplots for n and E are very similar. It appears that most subnetwork sizes are fairly small even for larger n or E values, so the subnetwork sizes may not increase with the network size or the number of edges. The maximum DP seems to be fairly large. The average DP is rather small for all networks regardless of their sizes. The boxplot for L/n indicates that most subnetworks represent less than 60% of the original network. The number of edges E is small in comparison to the total number of possible edges in the network due to small values of the quantity $E/n^2 \in [0, 0.13]$. This suggests that these networks do not have too many links.

parameters and that the ratio L/n decreases with increased n, E, which supports the observations from the boxplots.

Thus, the given data do not suggest a specific strong relationship between the numerical characteristics; however they allow us to observe trends and support some of the previous observations in the boxplots. Our samples are quite small, so it would be useful to continue adding new networks to the collection considered in this paper, to overcome the possible inaccuracies due to small sample size. The change of parameters in the network size algorithm leads to a fairly similar change in the subnetwork size for different networks as seen in **Figure 2**, suggesting a correlation between the choice of parameters and

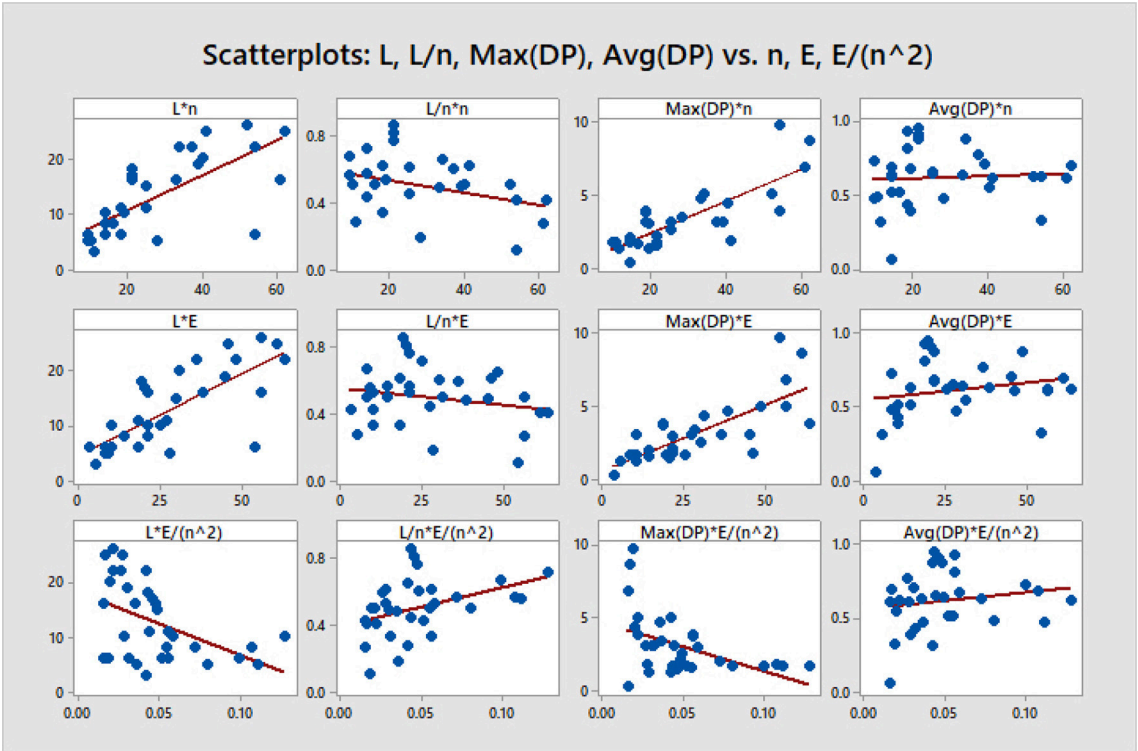


FIGURE 6 | Scatter plots and fitted lines for the identification of possible correlations between $L, L/n$, Max(DP), Avg(DP), and the parameters $n, E, E/n^2$. The equations for the fitted lines are listed in **Table 1**. There are no observable strong correlations and this is confirmed by the coefficients of determination in **Table 1**. Weak correlations are noticed for the increasing subnetwork size L as a function of n or E, and the increasing Max(DP) as a function of the same two parameters n and E. We also notice the decreasing trend of the ratio L/n with increased network size n or number of edges E, which suppports our observations from the boxplots.

TABLE 1 | Fitted lines and coefficients of determination R^2 corresponding to the scatter plots of **Figure 6**.

x \ y	L	L/n	Max(DP)	Avg(DP)
n	$y = 4.167 + 0.3187x$	$y = 0.6057 - 0.0037x$	$y = 0.176 + 0.1088x$	$y = 0.5906 + 0.0009x$
	$R^2 = 51.1\%$	$R^2 = 11.9\%$	$R^2 = 66.2\%$	$R^2 = 0.5\%$
E	$y = 4.638 + 0.2985x$	$y = 0.5622 - 0.002x$	$y = 0.6277 + 0.0912x$	$y = 0.5493 + 0.0024x$
	$R^2 = 56.9\%$	$R^2 = 4.7\%$	$R^2 = 59.1\%$	$R^2 = 4.8\%$
E/n^2	$y = 18.24 - 115x$	$y = 0.3961 + 2.315x$	$y = 4.697 - 33.29x$	$y = 0.5616 + 1.118x$
	$R^2 = 23\%$	$R^2 = 16.1\%$	$R^2 = 21.4\%$	$R^2 = 2.8\%$

The coefficients are generally small, the maximum values being observed for L vs. n, E and for Max(DP) vs. n, E. However, the maximum coefficient is only 66.2%, *which suggests weak correlations at best.*

the subnetwork size L. We expect that other possible variables or attributes that are intrinsic to the actual topology or dynamics of networks may have a stronger correlation with the DP values. Some of these attributes are connectivity (in-degree), number of outputs (out-degree), path length and other topological measures, canalizing depth, ratio of canalizing functions, or average bias of outputs (Albert and Barabasi, 2002; Kochi et al., 2014; Wohlgemuth and Matache, 2014). We plan on exploring them in great detail in future research to shed more light on possible relationships with the variables in **Figures 5**, **6**.

The observed general low DP values is what we expect in an equilibrium situation. It has been shown that the correlations between nodes become high only when facing a transition (Gorban et al., 2010; Censi and Calcagnini, 2011; Mojtahedi et al., 2016). It is possible that the simple node level hierarchy coming from mutual information might benefit from a study of at least some complex graph analysis descriptors such as in-degree, out-degree, betweenness and closeness centrality of the nodes that keep track of the role played by the nodes in the system they are embedded into (Csermely et al., 2005; Kovacs et al., 2010). In the next section we complement our analysis with a brief graph-theoretical perspective that is relevant in signaling networks (Di Paola and Giuliani, 2015).

4.2. Determinative Power and Topological Attributes

We will focus on some topological attributes or measures associated with the nodes of a BN that may provide more information on the magnitude of the DP values. Given a BN, $[n] := \{1, 2, \ldots, n\}$, and an arbitrary node $j \in [n]$, we consider the connectivity or the number k_j of inputs of the node j (the in-degree), the number o_j of outputs of the node j (the out-degree), together with several measures of centrality of node j as defined below.

DEFINITION 3. *A sequence of distinct nodes $P(i_1, i_m) = \{i_1, i_2, \ldots i_m\}$ of a BN with the property that i_k is an input to i_{k+1} for any $k = 1, 2, \ldots, m-1$, is called a path of length $m-1$ from the source node i_1 to the destination node i_m. Thus, the distance between the two nodes along this path is $d(i_1, i_m) = m - 1$.*

There could be multiple paths between two nodes, possibly with the same length. We are interested in the shortest path length between nodes. Observe that the shortest path length may differ if we switch the source and the destination nodes, so we may have $d(i_1, i_m) \neq d(i_m, i_1)$. On the other hand, if there is no path from node i to node j then $d(i, j) = 0$.

For a given node $i \in [n]$, let us consider the following quantities. We use the notation $|A|$ to denote the cardinality of the set A, in other words the number of elements in that set. Let

$$A_{in}(i) = |\{j \in [n] : j \neq i \text{ and there exists a path } P(j, i)\}|,$$
$$F_{in}(i) = \sum_{j \neq i} d(j, i),$$
$$A_{out}(i) = |\{j \in [n] : j \neq i \text{ and there exists a path } P(i, j)\}|,$$
$$F_{out}(i) = \sum_{j \neq i} d(i, j).$$

If $A_{in}(i) = 0$ then $F_{in}(i) = 0$, and similarly, if $A_{out}(i) = 0$ then $F_{out}(i) = 0$. The quantities $F_{in}(i), F_{out}(i)$ could be regarded as measures of the farness of node i from the other nodes in the network. The reciprocal of farness is a measure of closeness. If we multiply the closeness by the fraction of the sources or destinations of node i we obtain the following definitions of closeness centrality.

DEFINITION 4. *The in-closeness centrality of node $i \in [n]$ is the quantity*

$$C_{in}(i) = \left(\frac{A_{in}(i)}{N-1}\right)^2 \frac{1}{F_{in}(i)}, \qquad \text{if} \quad A_{in}(i) \neq 0,$$

and $C_{in}(i) = 0$ otherwise.

Similarly, the out-closeness centrality of node $i \in [n]$ is the quantity

$$C_{out}(i) = \left(\frac{A_{out}(i)}{N-1}\right)^2 \frac{1}{F_{out}(i)}, \qquad \text{if} \quad A_{out}(i) \neq 0,$$

and $C_{out}(i) = 0$ otherwise.

A second measure of centrality is the betweenness centrality, which measures how often each node appears on a shortest path between two nodes in the network. Given three distinct nodes i, j, k, let N_{jk} be the total number of shortest paths from j to k, and $N_{jk}(i)$ the number of those paths that pass through node i.

DEFINITION 5. *The betweenness centrality of node $i \in [n]$ is the quantity*

$$BC(i) = \sum_{j,k \neq i} \frac{N_{jk}(i)}{N_{jk}}.$$

The summation is over all nodes j, k for which $N_{jk} \neq 0$, meaning there exists at least a path between them.

We compute the topological attributes of nodes for the individual networks considered in previous figures, namely the *T-cell Receptor Signaling*, the *Oxidative Stress Pathway*, and the *Cholesterol Regulatory Pathway*. However, we are also adding one of the outlier networks, namely the signal transduction in fibroblast cells network with 130 nodes. The *Fibroblast Signaling* network has been investigated before in various publications (Kochi and Matache, 2012; Kochi et al., 2014; Matache and Matache, 2016; Puniya et al., 2016).

In **Figure 7** we provide network visualizations for each of the node attributes described in this section for the *Fibroblast Signaling* network. They are presented in the following order: DP, betweenness centrality, in-closeness centrality, out-closeness centrality, in-degree, and out-degree. The node color is proportional to the magnitude of these measures: dark colors for low values and light colors for large values. This type of visualization offers an overall view of the network's most central nodes, as well the nodes with most connections, or the nodes with highest DP values, thus identifying, to some extent, the role played by the nodes in the network they are embedded into. Similar graphs are shown in **Figure 8** for the *T-cell Receptor Signaling* network, in **Figure 9** for the *Oxidative Stress Pathway* network, and in **Figure 10** for the *Cholesterol Regulatory Pathway*.

We note that, aside from some similarities between the DP and the out-degree graphs which are expected given the definition of the DP as a summation of mutual information terms over all outputs of a given node, there is no other significant correlation. This is confirmed by a statistical analysis of the topological data. We include scatter plots with corresponding fitted regression lines for the DP as a function of the out-degree in **Figure 11**, together with the corresponding coefficients of determination R^2. The plots indicate that there might be nodes with high DP and fewer outputs, and also nodes with low DP and a larger number of outputs. In section 4.3 we relate this fact to the biological relevance of the nodes with large DP values. We provide simple scatter plots for DP as a function of the other topological measures indicating only the ranges of values of R^2 in **Figures 12–15**. The coefficient of variation is quite small in most cases. The largest values correspond to the DP vs. out-closeness and betweenness centrality of the smallest network, the *Oxidative Stress Pathway* network. However, even these values

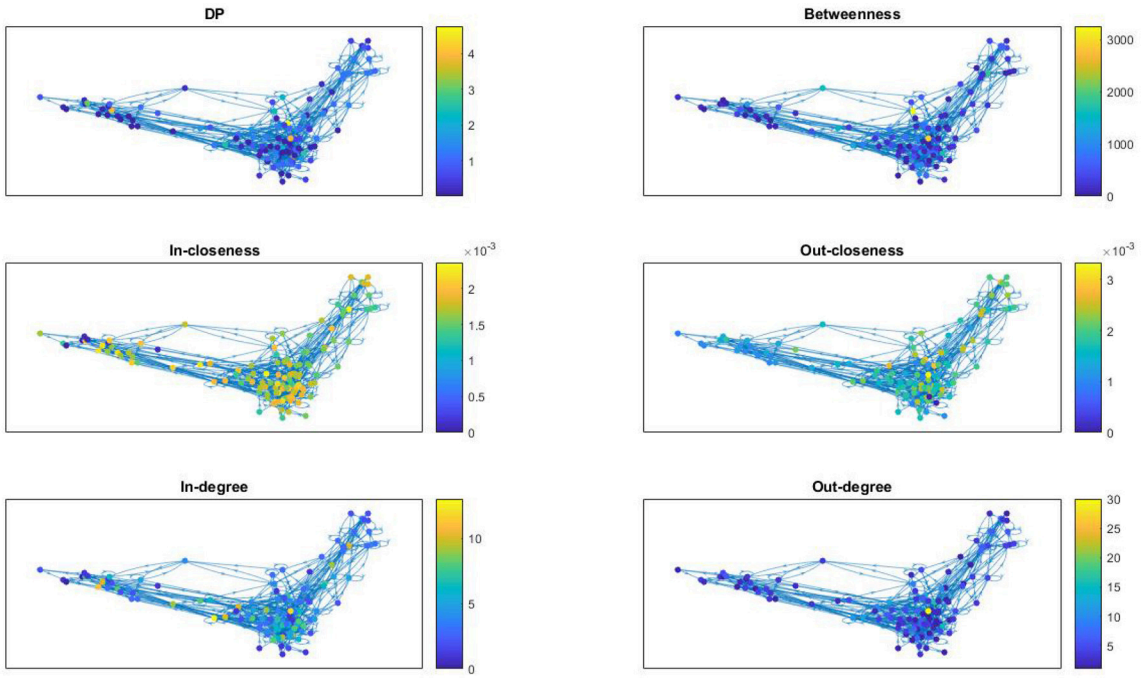


FIGURE 7 | *Fibroblast Signaling* network: visualization of node attributes.

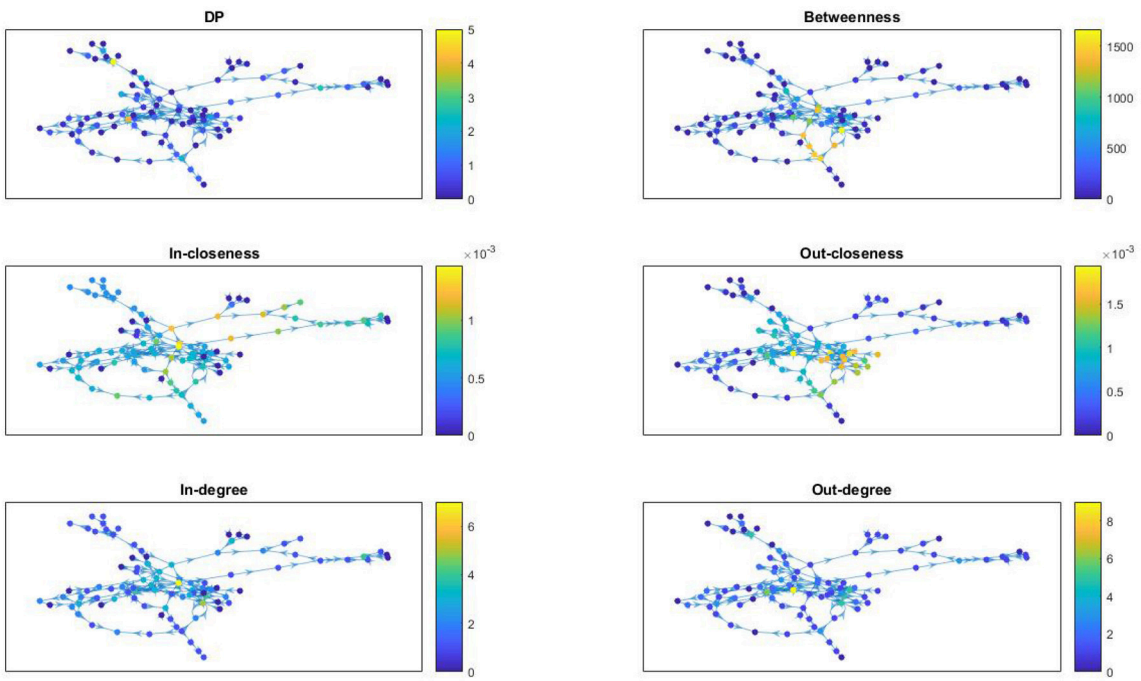


FIGURE 8 | *T-cell Receptor Signaling* network: visualization of node attributes.

are around 50%. We also conclude that for the four networks under consideration the DP is not correlated with any of the other topological measures.

Thus, further analyses need to be pursued, including other topological aspects in conjunction with various dynamical measures. For example, it has been shown that the location of

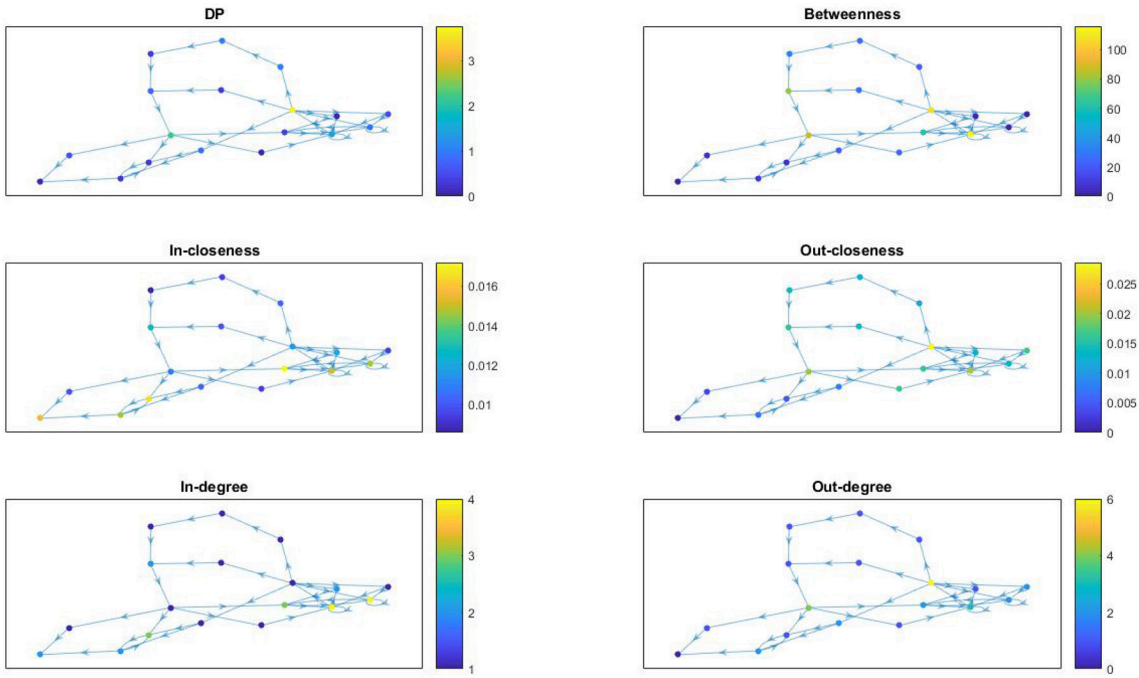


FIGURE 9 | *Oxidative Stress Pathway* network: visualization of node attributes.

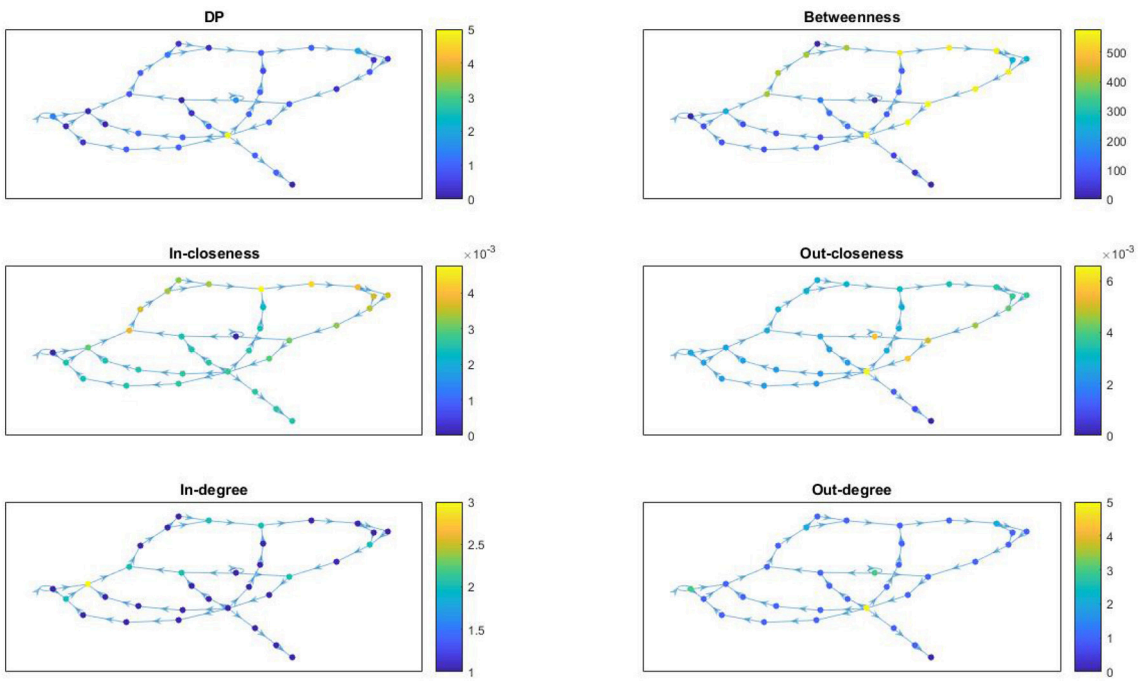


FIGURE 10 | *Cholesterol Regulatory Pathway* network: visualization of node attributes.

nodes in the network may be crucial for identifying enzymes whose elimination may have lethal effects in certain metabolic networks (Palumbo et al., 2005, 2007). In that case the metabolites are considered the nodes of the network, whereas the enzymes are the links between nodes. Therefore, it may be of further interest to explore other node location measures.

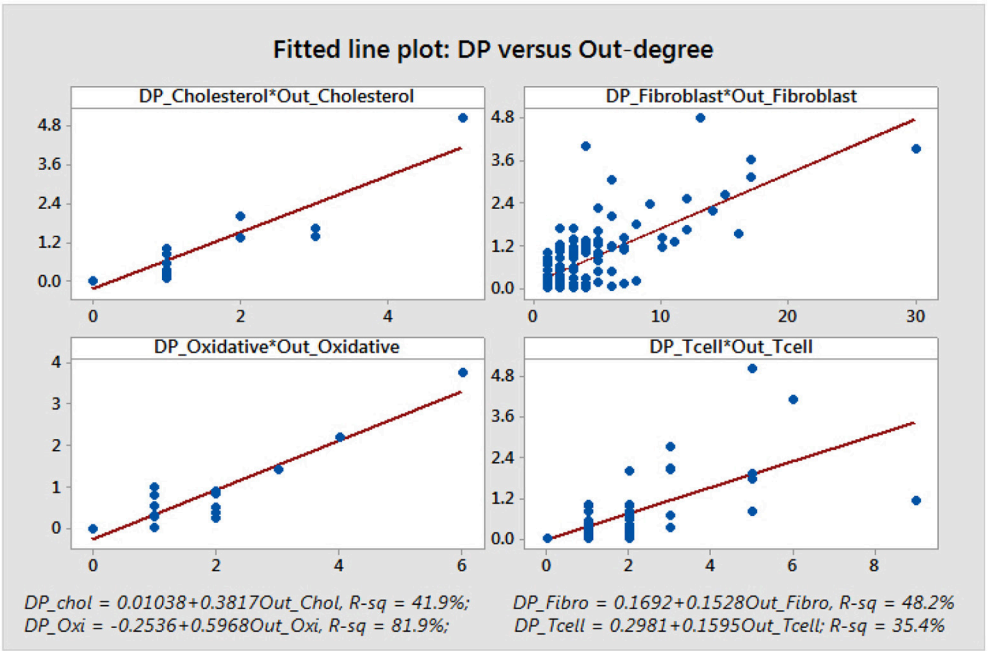


FIGURE 11 | The scatter plots suggest some correlation between the DP values and the number of outputs. However, we can observe that there might be situations where a large DP does not correspond to a large number of outputs. There can also be situations where the DP is small even though the node has more outputs.

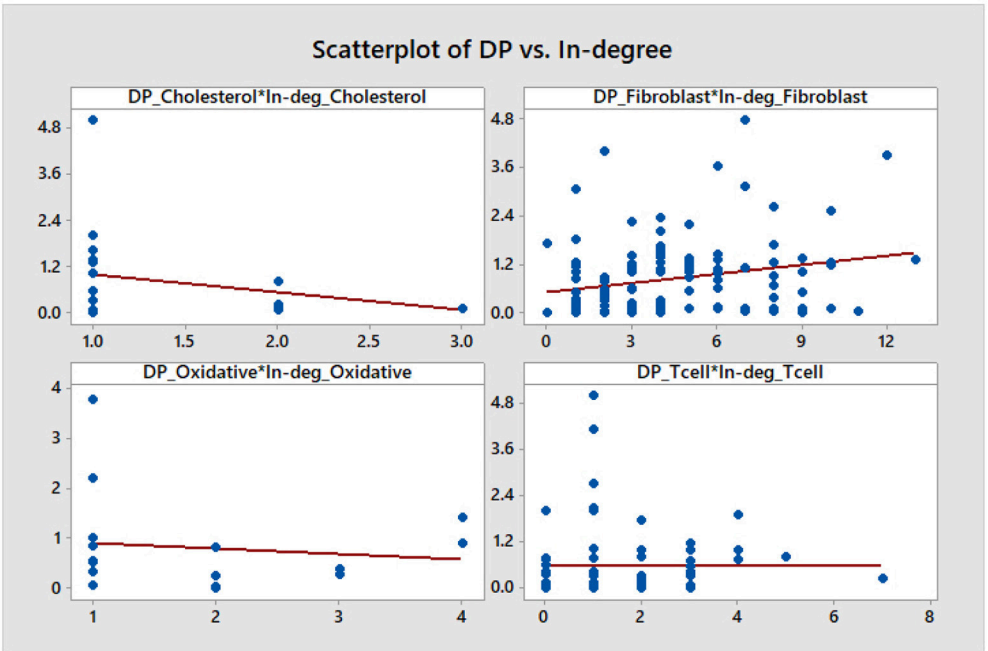


FIGURE 12 | Simple scatter plots for DP vs. in-degree. $R^2 \in [0\%, 7.3\%]$.

4.3. Biological Relevance of the Most Determinative Nodes

Aside from providing a method for finding a subnetwork with a fairly low impact on the overall entropy of the system, the DP method identifies biologically significant nodes among the top DP values. To support this statement we analyze biological relevance of the top DP nodes.

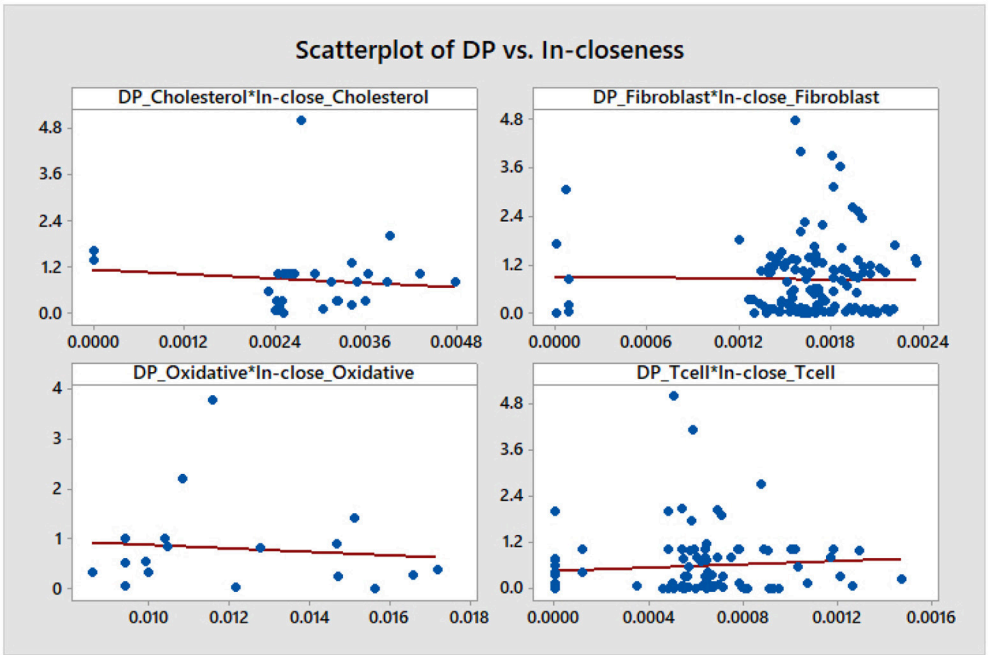


FIGURE 13 | Simple scatter plots for DP vs. in-closeness centrality. $R^2 \in [0\%, 1.3\%]$.

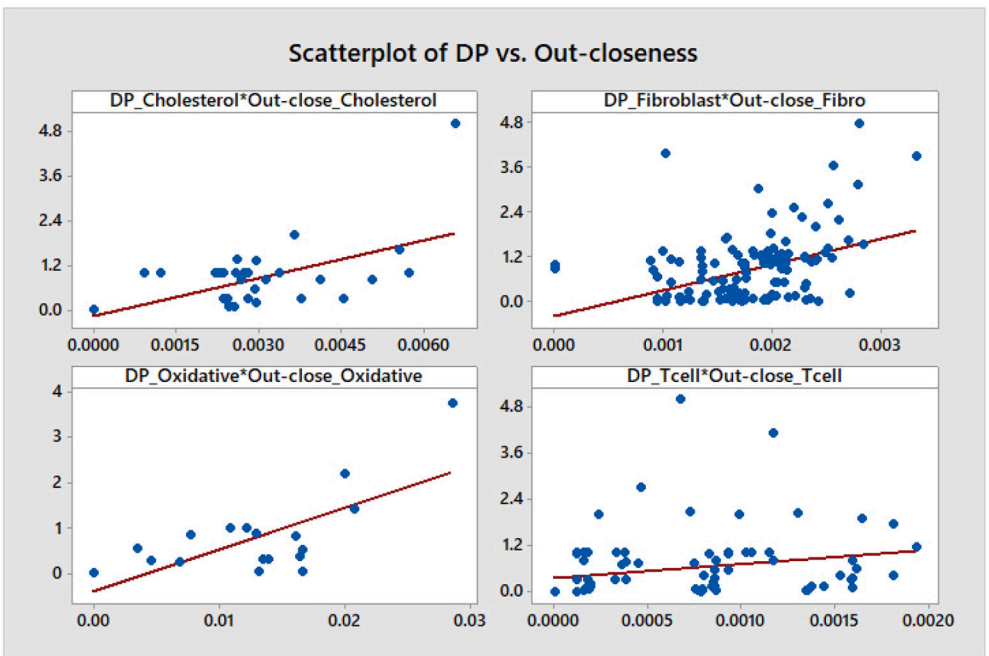


FIGURE 14 | Simple scatter plots for DP vs. out-closeness centrality. $R^2 \in [5.7\%, 48\%]$, where 48% corresponds to the *Oxidative Stress Pathway* network.

We focus on the particular networks shown in the figures so far, namely *Fibroblast Signaling*, *T-cell Receptor Signaling*, *Oxidative Stress Pathway*, *Cholesterol Regulatory Pathway*. These are all intercellular networks found in many different organisms. We are interested in the biological relationship between high DP and the nodes' biological importance in the cell. In our analysis, we provide most information on the larger networks from among these four, namely *Fibroblast Signaling* and *T-cell*

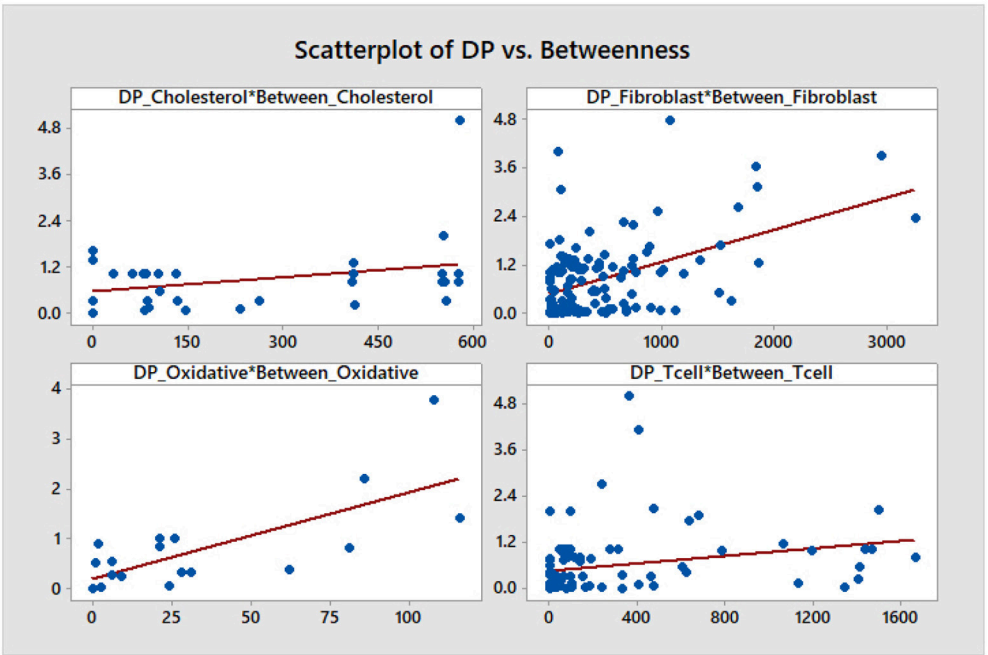


FIGURE 15 | Simple scatter plots for the DP vs. betweenness centrality. $R^2 \in [6.5\%, 52\%]$, where 52% corresponds to the *Oxidative Stress Pathway* network.

Receptor Signaling, and a shorter summary for the other two networks.

We start with a few notes on *Fibroblast Signaling*. To investigate in more detail whether nodes with high DP values are influential, we compare these nodes with the 32 most influential nodes identified under different environmental conditions in a previously published study by Puniya et al. (2016). We compare these most influential nodes with the top 10, 20, 30, 40, 50, and 60 nodes having high DP values in our analysis. We obtain an overlap of 70%, 65%, 50%, 47%, 38%, and 33%, respectively. Among the top 20 nodes having high DP values, 13 were previously identified as the most influential. Among the top 10, we find only one node which was previously identified as less influential. Similarly, in the top 20, 30, 40, 50, and 60 nodes, the distribution of the previously identified less influential nodes are 2, 3, 4, 8, and 13 respectively. This comparison suggests that the majority of nodes having high DP values ($>$ 65% in the top 20) are also identified as most influential when perturbed under different environmental conditions by Puniya et al. (2016). Therefore, these nodes may be involved in crucial biological functions.

Furthermore, we perform functional analyses of these nodes having high DP values. We provide information on all four networks under consideration.

1. **Methods**
   Gene essentiality data are obtained from the Online GEne Essentiality (OGEE) database version 1 that was downloaded on July 20, 2015 (Chen et al., 2012, 2017). Essential genes are deemed to be critical for cellular function and survival. As such, if an essential gene is removed (or knocked-out), it results in inviability. The OGEE database lists 7,168 genes as essential and 6,985 genes as conditionally (under specific environmental conditions) essential for humans, and was compiled using 18 different datasets of different cell lines using gene modification tools such as RNAi and CRISPER-cas9 (Chen et al., 2017). We overlap essential genes in that database with the nodes having high DP values in the *Fibroblast Signaling* network. Some nodes may be proteins that consist of multiple subunits or have multiple isoforms that are encoded by multiple genes. For example, Phospholipase D has two major isoforms, namely PLD 1 and PLD 2. Of these, PLD 1 is found to be essential in one tested cell line grown in GS-9 media (Chen et al., 2017). In such cases, we consider a node as essential if at least one gene (out of all protein coding genes) is listed as essential in the database. The proportion of the essential nodes in top selected nodes having high DP values is compared with the proportion of the essential nodes in the whole network. Using the DAVID tool for pathway enrichment analysis (Huang et al., 2009a,b), the genes associated with high DP nodes are mapped on the KEGG and Biocarta pathways and compared with the total genes in the network as a background. The DAVID tool uses Fisher's exact test to calculate p-values. The FDR is computed and a cutoff of 5% is used to correct the multiple comparisons. Furthermore, for annotation clustering the similar terms are clustered together using high classification stringency.

2. **Gene essentiality analysis**
   Fibroblast Signaling: To investigate the essentiality of the nodes with high DP values in the Fibroblast network, we map these nodes with gene essentiality data. Out of 130

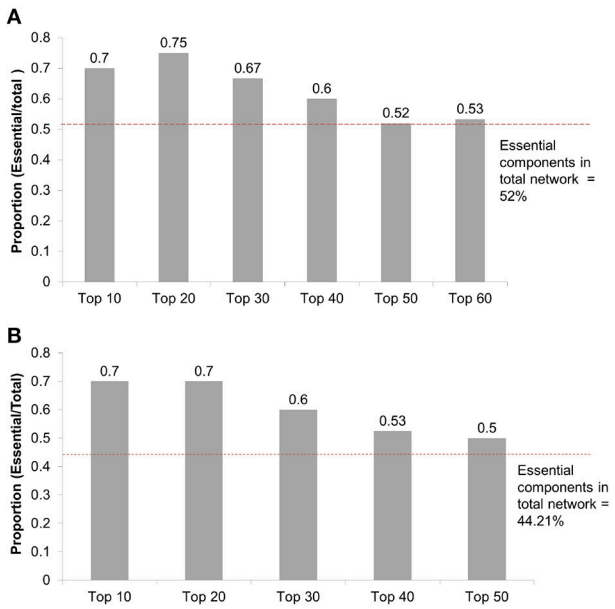


FIGURE 16 | (A) Distribution of essential nodes among top nodes having high DP values in the *Fibroblast Signaling* network with $n = 130$ nodes. Y-axis: Number of essential nodes among the selected nodes divided by number of selected nodes (top DP nodes). The red horizontal line indicates the proportion of essential nodes found in the whole network. **(B)** Distribution of essential nodes among top nodes having high DP values in the *T-cell Receptor Signaling* network with $n = 94$ nodes.

TABLE 2 | Essential genes among the Top 20 nodes having high DP values in the *Fibroblast Signaling* network.

Fibroblast Nodes (Top 20)	Essential Genes (Uniprot ID's)
ASK1	Q99683
CaM	Q96HY3
Cas	P56945
Cdc42	P60953
EGFR	Q504U8
Erk	P28482, Q8TD08, P27361, Q16659, P31152, Q13164, P53778
Fak	Q05397
IL1_TNFR	P01584, P19438
Mek	Q02750, P36507, P52564, P46734
PKA	P17612, P22694, P22612
PKC	P17252, P05771, P24723, Q05513, Q04759, Q02156, Q05655, P41743
PP2A	P67775
Rho	P08100
Src	P12931
Trafs	Q9BUZ4, Q9Y4K3

nodes in the network, 68 nodes (52%) are essential. To investigate the relationship between essentiality and DP values, we check the distribution of the essential nodes in the top 10, 20, 30, 40, 50, and 60 nodes having high DP values. The essential nodes in these top selected nodes are 70%, 75%, 66%, 60%, 52%, and 53% respectively, as shown in **Figure 16A**. High proportions of essential nodes are found in the top 10, 20, and 30 nodes. For the top 50 and 60 the proportions are close to the background proportion of essential nodes in the whole network. Among the top 20, a total of 15 nodes (75% of selected nodes) are identified as essential and are listed in **Table 2**. This proportion is significantly higher than the background proportion of essential nodes in the whole network (p-value $0.0306 < 0.05$).

T-cell Receptor Signaling: We investigate the distribution of essential genes in T-cell signaling model. A total of 42 nodes out of 95 (42.2%) are essential. Among the top 10, 20, 30, 40, and 50 nodes having high DP values, 7, 14, 18, 21, and 25 are essential as shown in **Figure 16B**. We find 70% of nodes as essential in each of the top 10 and top 20 nodes. The proportion of the essential nodes decreases with decreasing DP value. The proportion of the essential nodes in the top 20 nodes having high DP values is significantly higher than that of the background proportion of 42.2% in the whole network (p-value $0.0115 < 0.05$).

Oxidative Stress Pathway: Oxidative stress signaling model consists of 18 nodes. Of these, 13 nodes (72.22%) are essential. In the top 5 and top 10 nodes having high DP values, 4 and 7 are essential, respectively. For example, the top hub nodes ROS and AKT are essential.

Cholesterol Regulatory Pathway: Out of 34 nodes, 7 are essential. The top hub node msREBP is essential in metabolic reprogramming of the effector T-cells (Kidani et al., 2013).

Thus, nodes having high DP values are enriched with essential genes suggesting that the DP values might be used to predict the gene or protein essentiality.

We include here a note on how the gene essentiality results relate to the cutoff L for the subnetwork size. For example for the *T-cell Receptor Signaling* network shown in **Figure 16B**, we find 53% essential nodes among the top $L = 40$ nodes having high DP values, in comparison to the 44% essential nodes in the whole network. Similarly, for the *Cholesterol Regulatory Pathway* network a total of 7 essential genes (20%) are found. Of these, 5 nodes are in the top $L = 22$ nodes having high DP. Furthermore, in the case of the *Oxidative Stress Pathway* network, we find 5 essential nodes out of $L = 6$ nodes compared to 13 out of the 18 in whole network. Thus, our chosen cutoff L seems to be sufficient for identifying a large fraction of essential nodes. Moreover, the results suggest that even smaller values of the cutoff L would allow a significant identification of essential nodes.

3. **Biological pathway analysis**

Fibroblast Signaling: Further, to investigate the biological processes associated with top DP nodes, we perform pathway analysis of nodes having high DP values (Top 20). We obtain 15 KEGG pathways including signaling pathways such as TNF-alpha signaling, MAPK signaling, and

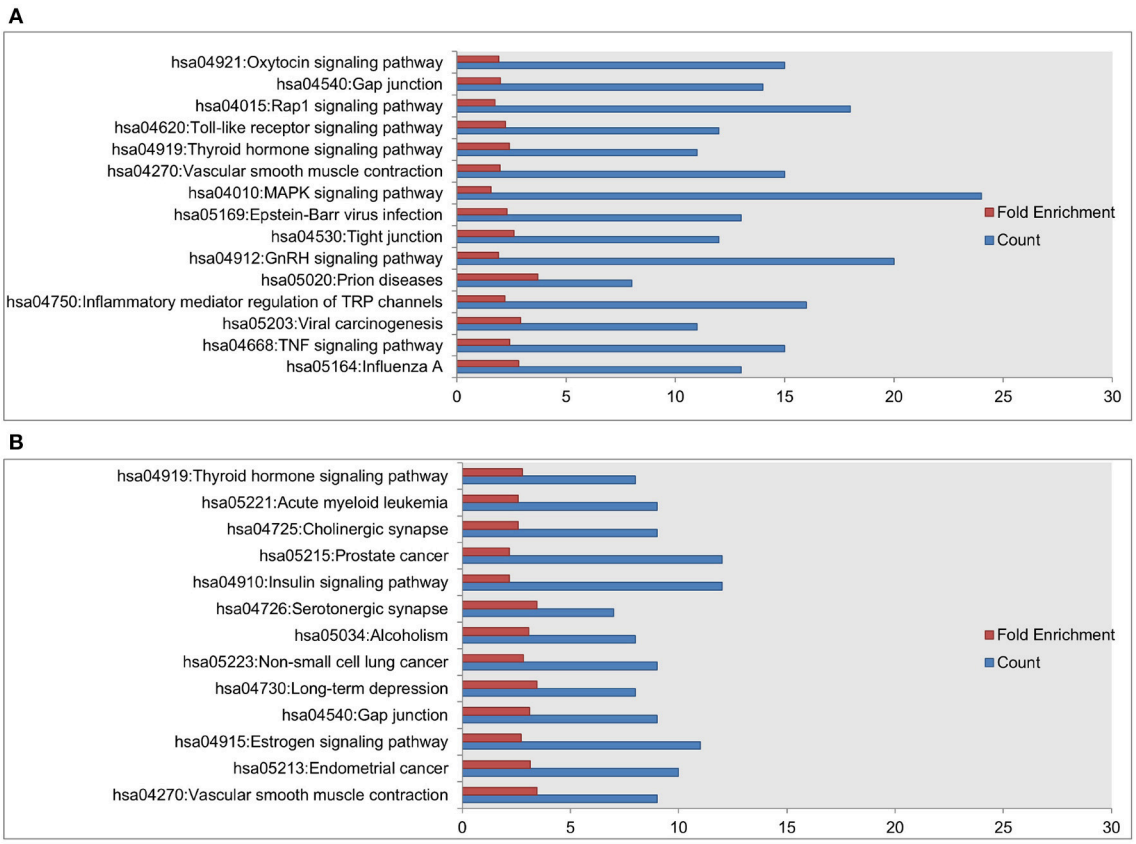


FIGURE 17 | Enriched KEGG pathways among the top 20 selected nodes having high DP values in the **(A)** *Fibroblast Signaling* network with 130 nodes and **(B)** *T-cell Receptor Signaling* network with 94 nodes. The values on the x-axis correspond to fold enrichment and the total number of genes found in the KEGG pathway. The enriched pathways are given on the y-axis.

TLR signaling, and pathways associated with diseases such as influenza A infection, viral carcinogenesis, prion diseases, and Epstein-Barr virus infection. The results are shown in **Figure 17A**. The Erk node is common among 14 out of 15 enriched pathways. Next to this, the Mek node is common among 13 out of 15 enriched KEGG pathways. The EGFR node that has the highest DP value is involved in 5 KEGG pathways. These results suggest that the nodes having high DP values are involved in crucial biological functions, and are also associated with a variety of infections and diseases.

T-cell Receptor Signaling: Among the top 20 nodes having high DP values, we obtain 13 enriched KEGG pathways as seen in **Figure 17B**. These enriched pathways include insulin signaling, and pathways involved in diseases such as cancers, long term depression, and alcoholism. The node Raf is common among 12 out of 13 enriched KEGG pathways. The pkb node has the highest DP value in the *T-cell Receptor Signaling* network and is involved in 8 out of 13 enriched KEGG pathways. These results suggest that the nodes having high DP values are involved in crucial biological functions, and also associated with a variety of diseases including cancers.

Oxidative Stress Pathway: Among the top 5 nodes, the KEGG pathways including renal cell carcinoma, acute myeloid leukemia, prolactin, estrogen, B-cell receptor, and the T-cell receptor are found to be enriched.

Cholesterol Regulatory Pathway: Among the top 20 nodes no KEGG pathway is found to be enriched.

5. DISCUSSION

The biological function analysis of the nodes having high DP values (hubs) in the *Fibroblast Signaling*, *T-cell Receptor Signaling*, *Oxidative Stress Pathway*, and *Cholesterol Regulatory Pathway* networks suggest that the majority of nodes are essential and also involved in crucial biological functions. The proportion of the essential nodes among nodes having high DP values (e.g., top 20) in large scale models, i.e., *Fibroblast Signaling* (130 nodes) and *T-cell Receptor Signaling* (94 nodes) is significantly higher than that of the total

essential nodes in the whole network. On the other hand, the comparatively small models *Oxidative Stress Pathway* and *Cholesterol Regulatory Pathway* models also exhibit their hub nodes as essential. The biological pathway analysis of top hub nodes shows that these are involved in important disease pathways.

To have a better understanding of the meaning of the subnetworks of hubs in the more general context of the whole networks, we provide further insight into the biological roles of some of the top hubs in each of the four networks.

The *Fibroblast Signaling* network is a generic network that consists of several major signaling pathways including the Epidermal Growth Factor Receptor (EGFR), the G-protein coupled receptor, and the integrin signaling pathway (Puniya et al., 2016). In the *Fibroblast Signaling* network the nodes with the highest DP values e.g., EGFR, Apoptosis signal-regulationg kinase 1 (ASK1), Erk, Focal adhesion kinase (Fak), Cellular apoptosis susceptibility (Cas) protein, Calmodulin (CaM), or Mek have critical functions in the protein kinase activity, the regulation of protein kinase activity, and the cell proliferation and apoptosis. For example, the hub node EGFR is found to be essential for several biological functions, such as in Toll-like Receptor 3 signaling in human and mouse cell types, including fibroblast, dendritic cells, and macrophages (Yamashita et al., 2012).

The *T-cell Receptor Signaling* network comprises the T-cell receptor, its co-receptors and the transcription factors involved in T-cell activation and function (Saez-Rodriguez et al., 2007). In this network, the nodes with the highest DP values include Protein Kinase B (pkb), Linker of Activated T-cells (Lat), Fyn, Zap70, and the tyrosine kinase (lckp1), that have important roles in the T-cell receptor signaling. The hub node Zap70 is a tyrosine kinase that is essential for the adaptive immune response (Wang et al., 2010). Furthermore, the protein associated with the Lat node is phosphorylated by Zap70 following the T-cell receptor activation (Paz et al., 2001). The other nodes, i.e., pkb, Fyn, and lckp1, are tyrosine kinases involved in cell growth and proliferation (Safran et al., 2010).

The *Oxidative Stress Pathway* network comprises the oxidative stress and PI3K/Akt signaling. In this network, the nodes reactive oxygen species (ROS), Akt and the Anti-oxidant response element (ARE) have the highest DP values. ROS plays an important role in the maintenance of the redox balance. Increased levels of ROS causes macromolecules and cell organelle damage, and triggers the cell apoptosis (Redza-Dutordoir and Averill-Bates, 2016). On the other hand Akt is a positive regulator of cell proliferation.

The *Cholesterol Regulatory Pathway* network consists of reactions involved in cholesterol biosynthesis and its regulation by Sterol regulatory element-binding proteins (SERBPs). The nodes with the highest DP values include mSREBP, Statins, and Acetyl-CoA, and have important roles in regulation. The node mSREBP is a transcription activator involved in the lipid biosynthesis pathway (Shimano, 2001). The Statins are inhibitors of cholesterol biosynthesis. The Acetyl-CoA is a central metabolite and a substrate for cholesterol biosynthesis.

We also point out that many essential nodes may tend to have a large number of outputs, and since the DP is a summation of MI values over all possible outputs, there is a natural correlation between higher DP values and larger number of outputs, as noted in Matache and Matache (2016) and as seen in **Figure 11**. However, the DP method can identify essential nodes with both large and small number of outputs.

For example, in the *Fibroblast Signaling* network, the top DP node is *EGFR* having 13 outputs. It is identified as an essential node. In Matache and Matache (2016) it is specified that mutations of the *EGFR* are known to be related to lung cancer, interfering with the signaling pathways within the cell triggered to promote cell growth and division (proliferation) and cell survival. The second node in the order of DP is *ASK1*, also an essential node. This node has only 4 outputs and plays important roles in many stress-related diseases, including cancer, diabetes, cardiovascular, and neurodegenerative diseases. The third is the proto-oncogene tyrosine-protein kinase (*Src*), identified as essential. This node is involved in the control of many functions, including cell adhesion, growth, movement and differentiation, and has 30 outputs. Although the fourth node Phosphatidylinositol (3,4,5)-trisphosphate (*PIP3_345*) has 17 outputs, it is not considered essential in the OGEE database (Chen et al., 2012). In fact, among the top 20% of nodes with large DP values, we identify as essential 80% of those with large ($\geq$ 6) number of outputs and 50% of those with small ($\leq$ 5) number of outputs. The average number of outputs is 4.3 and the maximum is 30 in the *Fibroblast Signaling* network.

A fairly similar situation occurs for the *T-cell Receptor Signaling* network. This suggests that future studies will need to look at further correlations between essentiality and DP values.

We note here that the codes used for the work in this paper are available upon request.

6. CONCLUSIONS

Our results suggest that DP can serve as a useful tool to identify a subset of relevant nodes in the network that offer the most information gain and whose knowledge reduces the entropy of the whole network significantly. Moreover, many of the nodes with top DP values are identified as biologically essential.

Several directions for further research include extending the data to other networks to increase our samples for the statistical analysis, as well as identifying some network properties or attributes that are potentially correlated with the DP values, such as average bias of the outputs of nodes, canalizing depth, clustering coefficients, or feedback loop information. Moreover, most biological networks have a very large maximal strongly connected component called the "core" (Steinway et al., 2015; Gan and Albert, 2016). On the other hand, it has been shown that disrupting nodes that do not belong to the core may have a significant impact on the network (Palumbo et al., 2005, 2007). More precisely, essential mutations corresponding to enzymes whose elimination has lethal effects on a metabolic network, tend

to have a peripheral position and are seldom located in highly connected components of the network. It would be of interest to know how the DP values in the core differ from those not in the core to possibly unravel further correlations.

Another topic for further research is to perform actual network reduction to its top DP nodes and compare the dynamics of the subnetwork to the dynamics of the entire network to explore further the ability of the subnetwork to capture important dynamical aspects of the whole network, such as preservation of attractors. For instance, it would be of interest to explore the Java software GINsim (Naldi et al., 2009a) to actually perform the network reduction and use it to analyze dynamics of the various models found in Cell Collective. This endeavor will require a suitable algorithm for eliminating the edges or connections linking the nodes of the chosen subnetwork to the eliminated nodes.

Some more theoretical approaches would be to study the impact of network reduction for homogeneous networks (that is, networks in which all nodes obey a certain type of Boolean function) to set some baseline dynamical behavior to be used for comparison with more realistic network models.

AUTHOR CONTRIBUTIONS

TP developed some of the computer codes needed for data collection from the Cell Collective, and he performed most of the simulations needed to generate the data related to the DP values and the subnetwork size for all networks under discussion. He also wrote some of the related parts of the manuscript. BP applied the selected methods for analyzing the biological relevance of the most determinative nodes and wrote the related parts of the manuscript. TH selected the most suitable methods for analyzing the biological relevance of the most determinative nodes, and provided support with network selection, accuracy of approach, and biological information. He wrote the related parts of the manuscript and formatted it for submission. MM devised the mathematical method for finding the subnetwork size, the computer codes for calculations of the DP values and the subnetwork size, and she performed the statistical analysis. She wrote the related parts of the manuscript and formatted it for submission. All authors have contributed to the revision of the manuscript and have agreed on the final draft.

FUNDING

TP was supported by the University of Nebraska-Omaha FUSE and KRMP student grants for 2015-2016, while TH was supported by NIH grant no. 5R35GM119770-02.

ACKNOWLEDGMENTS

We are grateful for the discussions related to the biological relevance of the DP we had with Dr. Jim Rogers from the University of Nebraska at Omaha.

REFERENCES

Abou-Jaoudé, W., Monteiro, P. T., Naldi, A., Grandclaudon, M., Soumelis, V., Chaouiya, C., et al. (2015). Model checking to assess t-helper cell plasticity. *Front. Bioeng. Biotechnol.* 2:86. doi: 10.3389/fbioe.2014.00086

Abou-Jaoudé, W., Traynard, P., Monteiro, P. T., Saez-Rodriguez, J., Helikar, T., Thieffry, D., et al. (2016). Logical modeling and dynamical analysis of cellular networks. *Front. Genet.* 7:94. doi: 10.3389/fgene.2016.00094

Albert, R., and Barabasi, A.-L. (2002). Statistical mechanics of complex networks. *Mod. Phys.* 74, 47–97. doi: 10.1103/RevModPhys.74.47

Albert, R., and Othmer, H. (2003). The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*. *J. Theor. Biol.* 223, 1–18. doi: 10.1016/S0022-5193(03)00035-3

Albert, R., and Thakar, J. (2014). Boolean modeling: a logic-based dynamic approach for understanding signaling and regulatory networks and for making useful predictions. *Wiley Interdiscip. Rev.* 6, 353–369. doi: 10.1002/wsbm.1273

Bilke, S., and Sjunnesson, F. (2001). Stability of the kauffman model. *Phys. Rev.* 65:016129. doi: 10.1103/PhysRevE.65.016129

Censi, F., Giuliani, A., Bartolini, P., and Calcagnini, G. (2011). A multiscale graph theoretical approach to gene regulation networks: a case study in atrial fibrillation. *IEEE Trans. Biomed. Eng.* 10, 2943–2946. doi: 10.1109/TBME.2011.215074

Chen, W. H., Lu, G., Chen, X., Zhao, X. M., and Bork, P. (2017). Ogee v2: an update of the online gene essentiality database with special focus on differentially essential genes in human cancer cell lines. *Nucleic Acids Res.* 45, D940–D944. doi: 10.1093/nar/gkw1013

Chen, W. H., Minguez, P., Lercher, M. J., and Bork, P. (2012). Ogee: an online gene essentiality database. *Nucleic Acids Res.* 40, D901–D906. doi: 10.1093/nar/gkr986

Conroy, B. D., Herek, T. A., Shew, T. D., Latner, M., Larson, J. J., Allen, L., et al. (2014). Design, assessment, and *in vivo* evaluation of a computational model illustrating the role of cav1 in cd4(+) t-lymphocytes. *Front. Immunol.* 5:599. doi: 10.3389/fimmu.2014.00599

Cover, T. M., and Thomas, J. A. (2006). *Elements of Information Theory*. Hoboken, NJ: Wiley-Interscience.

Csermely, P., Agoston, V., and Pongor, S. (2005). The efficiency of multi-target drugs: the network approach might help drug design. *Trends Pharmacol. Sci.* 4, 178–182. doi: 10.1016/j.tips.2005.02.007

Di Paola, L., and Giuliani, A. (2015). Protein contact network topology: a natural language for allostery. *Curr. Opin. Struct. Biol.* 31, 43–48. doi: 10.1016/j.sbi.2015.03.001

Gan, X., and Albert, R. (2016). Analysis of a dynamic model of guard cell signaling reveals the stability of signal propagation. *BMC Syst. Biol.* 10:78. doi: 10.1186/s12918-016-0327-7

Gorban, A. N., Smirnova, E. V. and Tyukina, T. A. (2010). Correlations, risk and crisis: from physiology to finance. *Physica A* 16, 3193–3217. doi: 10.1016/j.physa.2010.03.035

Heckel, R., Schober, S., and Bossert, M. (2013). Harmonic analysis of boolean networks: determinative power and perturbations. *EURASIP J. Bioinform. Syst. Biol.* 2013:6. doi: 10.1186/1687-4153-2013-6

Helikar, T., Konvalina, J., Heidel, J., and Rogers, J. A. (2008). Emergent decision-making in biological signal transduction networks. *Proc. Natl. Acad. Sci. U.S.A.* 105, 1913–1918. doi: 10.1073/pnas.0705088105

Helikar, T., Kowal, B., McClenathan, S., Bruckner, M., Rowley, T., Madrahimov, A., et al. (2012). The cell collective: toward an open and collaborative approach to systems biology. *BMC Syst. Biol.* 6:96. doi: 10.1186/1752-0509-6-96

Helikar, T., Kowal, B., and Rogers, J. A. (2013). A cell simulator platform: the cell collective. *Clin. Pharmacol. Ther.* 93, 393–395. doi: 10.1038/clpt.2013.41

Huang da, W., Sherman, B. T., and Lempicki, R. A. (2009a). Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Res.* 37, 1–13. doi: 10.1093/nar/gkn923

Huang da, W., Sherman, B. T., and Lempicki, R. A. (2009b). Systematic and integrative analysis of large gene lists using david bioinformatics resources. *Nat. Protoc.* 4, 44–57. doi: 10.1038/nprot.2008.211

Irons, D. J. (2009). Logical analysis of the budding yeast cell cycle. *J. Theor. Biol.* :4. doi: 10.1016/j.jtbi.2008.12.028

Kauffman, S. A. (1993). *The Origins of Order*. New York, NY: Oxford University Press, 173–235.

Kaufman, V., and Drossel, B. (2006). Relevant components in critical random boolean networks. *New J. Phys.* 8:228. doi: 10.1088/1367-2630/8/10/228

Kaufman, V., Mihaljev, T., and Drossel, B. (2005). Scaling in critical random boolean networks. *Phys. Rev.* 72:4. doi: 10.1103/PhysRevE.72.046124

Kervizic, G., and Corcos, L. (2008). Dynamical modeling of the cholesterol regulatory pathway with boolean networks. *BMC Syst. Biol.* 2:99. doi: 10.1186/1752-0509-2-99

Kidani, Y., Elsaesser, H., Hock, M. B., Vergnes, L., Williams, K. J., Argus, J. P., et al. (2013). Sterol regulatory element—binding proteins are essential for the metabolic programming of effector t cells and adaptive immunity. *Nat. Immunol.* 14, 489–499. doi: 10.1038/ni.2570

Klemm, K., and Bornholdt, S. (2000). Stable and unstable attractors in boolean networks. *Phys. Rev.* 72:055101. doi: 10.1103/PhysRevE.72.055101

Kochi, N., Helikar, T., Allen, L., Rogers, J. A., Wang, Z., and Matache, M. T. (2014). Sensitivity analysis of biological boolean networks using information fusion based on nonadditive set functions. *BMC Syst. Biol.* 8:92. doi: 10.1186/s12918-014-0092-4

Kochi, N., and Matache, M. T. (2012). Mean-field boolean network model of a signal transduction network. *Biosystems* 108, 14–27. doi: 10.1016/j.biosystems.2011.12.001

Kovacs, I. A., Palotai, R., Szalay, M. S., and Csermely, P. (2010). Community landscapes: an integrative approach to determine overlapping network module hierarchy, identify key nodes and predict network dynamics. *PLoS ONE* 9:e12528. doi: 10.1371/journal.pone.0012528

Krawitz, P., and Shmulevich, I. (2007a). Basin entropy in boolean network ensembles. *Phys. Rev. Lett.* 98:158701. doi: 10.1103/PhysRevLett.98.158701

Krawitz, P., and Shmulevich, I. (2007b). Entropy of complex relevant components of boolean networks. *Phys. Rev. E* 76:036115. doi: 10.1103/PhysRevE.76.036115

Marques-Pita, M., and Rocha, L. M. (2013). Canalization and control in automata networks: body segmentation in *Drosophila melanogaster*. *PLoS ONE* 8:e55946. doi: 10.1371/journal.pone.0055946

Matache, M. T., and Matache, V. (2016). Logical reduction of biological networks to their most determinative components. *Bull. Math. Biol.* 78, 1520–1545. doi: 10.1007/s11538-016-0193-x

Mendéz, A., and Mendoza, L. (2016). A network model to describe the terminal differentiation of b cells. *PLoS Comput. Biol.* 12:e1004696. doi: 10.1371/journal.pcbi.1004696.

Mojtahedi, M., Skupin, A., Zhou, J., Castano, I. G., Leong-Quong, R. Y., Chang, H.,et al . (2016). Cell fate decision as high-dimensional critical state transition. *PLoS Biol.* 12:e2000640. doi: 10.1371/journal.pbio.2000640

Naldi, A., Berenguier, D., Fauré, A., Lopez, F., Thieffry, D., and Chaouiya, C. (2009a). Logical modelling of regulatory networks with ginsim 2.3. *Biosystems* 97, 134–139. doi: 10.1016/j.biosystems.2009.04.008

Naldi, A., Remy, E., Thieffry, D., and Chaouiya, C. (2009b). A reduction of logical regulatory graphs preserving essential dynamical properties. *Comput. Methods Syst. Biol.* 5688, 266–280. doi: 10.1007/978-3-642-03845-7_18

Palumbo, M.C., Colosimo, A., Giuliani, A., and Farina, L. (2005). Functional essentiality from topology features in metabolic networks: a case study in yeast. *FEBS Lett.* 21, 4642–4646. doi: 10.1016/j.febslet.2005.07.033

Palumbo, M.C., Colosimo, A., Giuliani, A., and Farina, L. (2007). Essentiality is an emergent property of metabolic network wiring. *FEBS Lett.* 13, 2485–2489. doi: 10.1016/j.febslet.2007.04.067

Paz, P. E., Wang, S., Clarke, H., Lu, X., Stokoe, D., and Abo, A. (2001). Mapping the zap-70 phosphorylation sites on lat (linker for activation of t cells) required for recruitment and activation of signalling proteins in t cells. *Biochem J.* 356, 461–471. doi: 10.1042/bj3560461

Puniya, B., Allen, L., Hochfelder, C., Majumder, M., and Helikar, T. (2016). Systems perturbation analysis of a large-scale signal transduction modelreveals potentially influential candidates for cancer therapeutics. *Front. Bioeng. Biotechnol.* 11:10. doi: 10.3389/fbioe.2016.00010

Redza-Dutordoir, M., and Averill-Bates, D. (2016). Activation of apoptosis signalling pathways by reactive oxygen species. *Biochim. Biophys. Acta* 1863, 2977–2992. doi: 10.1016/j.bbamcr.2016.09.012

Ribeiro, A. S., Kauffman, S. A., Lloyd-Price, J., Samuelsson, B., and Socolar, J. E. (2008). Mutual information in random boolean models of regulatory networks. *Phys. Rev. E* 77:011901. doi: 10.1103/PhysRevE.77.011901

Richardson, K. A. (2004). Simplifying boolean networks. *Adv. Complex Syst.* 8, 365–381. doi: 10.1142/S0219525905000518

Saadatpour, A., Albert, R., and Reluga, T. C. (2013). A reduction method for boolean network models proven to conserve attractors. *SIAM J. Appl. Dyn. Syst.* 12, 1997–2011. doi: 10.1137/13090537X

Saez-Rodriguez, J., Simeoni, L., Lindquist, J. A., Hemenway, R., Bommhardt, U., Arndt, B., et al. (2007). A logical model provides insights into t cell receptor signaling. *PLoS Comput. Biol.* 3:e163. doi: 10.1371/journal.pcbi.0030163

Safran, M., Dalah, I., Alexander, J., Rosen, N., Iny Stein, T., Shmoish, M., et al. (2010). Genecards version 3: the human gene integrator. *Database* 2010:baq020. doi: 10.1093/database/baq020

Shimano, H. (2001). Sterol regulatory element-binding proteins (srebps): transcriptional regulators of lipid synthetic genes. *Prog. Lipid Res.* 40, 439–452. doi: 10.1016/S0163-7827(01)00010-8

Shmulevich, I., Dougherty, E. R., and Zhang, W. (2002). "From boolean to probabilistic boolean networks as models for genetic regulatory networks," in *Proceedings of the IEEE*, 1778–1792.

Shmulevich, I., and Kauffman, S. A. (2004). Activities and sensitivities in boolean network models. *Phys. Rev. Lett.* 93:048701. doi: 10.1103/PhysRevLett.93.048701

Socolar, J. E., and Kauffman, S. A. (2003). Scaling in ordered and critical random boolean networks. *Phys. Rev.* 90:068702. doi: 10.1103/PhysRevLett.90.068702

Sridharan, S., Layek, R., Datta, A., and Venkatraj, J. (2012). Boolean modeling and fault diagnosis in oxidative stress response. *BMC Genomics* 13:S4. doi: 10.1186/1471-2164-13-S6-S4

Steinway, S. N., Biggs, M. B., Loughran, T. P., Papin, J. A., and Albert, R. (2015). Inference of network dynamics and metabolic interactions in the gut microbiome. *PLoS Comput. Biol.* 11:e1004338. doi: 10.1371/journal.pcbi.1004338.

Todd, R. G., and Helikar, T. (2012). Ergodic sets as cell phenotype of budding yeast cell cycle. *PLoS ONE* 7:e45780. doi: 10.1371/journal.pone.0045780

Veliz-Cuba, A. (2011). Reduction of boolean network models. *J. Theor. Biol.* 289, 167–172. doi: 10.1016/j.jtbi.2011.08.042

Wang, H., Kadlecek, T. A., Au-Yeung, B. B., Goodfellow, H. E., Hsu, L. Y., Freedman, T. S., et al. (2010). Zap-70: An essential kinase in t-cell signaling. *Cold Spring Harb. Perspect. Biol.* 2:a002279. doi: 10.1101/cshperspect.a002279

Wohlgemuth, J., and Matache, M. T. (2014). Small world properties of facebook group networks. *Complex Syst.* 23, 197–225.

Yamashita, M., Chattopadhyay, S., Fensterl, V., Saikia, P., Wetzel, J., and Sen, G. (2012). Epidermal growth factor receptor is essential for toll-like receptor 3 signaling. *Sci. Signal.* 5:ra50. doi: 10.1126/scisignal.2002581

Conflict of Interest Statement: TH has served as a shareholder and/or has consulted for Discovery Collective, Inc.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

ORIGINAL RESEARCH
published: 07 September 2018
doi: 10.3389/fphys.2018.01161

Estimating Attractor Reachability in Asynchronous Logical Models

Nuno D. Mendes[1†], *Rui Henriques*[2,3†], *Elisabeth Remy*[4], *Jorge Carneiro*[1], *Pedro T. Monteiro*[2,3*] *and Claudine Chaouiya*[1*]

[1] *Instituto Gulbenkian de Ciência, Oeiras, Portugal,* [2] *Department of Computer Science and Engineering, Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal,* [3] *Instituto de Engenharia de Sistemas e Computadores Investigação e Desenvolvimento, Lisbon, Portugal,* [4] *Aix Marseille University, CNRS, Centrale Marseille, I2M UMR 7373, Marseille, France*

OPEN ACCESS

Edited by:
Xiaogang Wu,
University of Nevada, Las Vegas, United States

Reviewed by:
Jeffrey Varner,
Purdue University, United States
Brandilyn Stigler,
Southern Methodist University, United States
Benjamin Andrew Hall,
University of Cambridge, United Kingdom
Arnau Montagud,
Institut Curie, France

***Correspondence:**
Pedro T. Monteiro
Pedro.Tiago.Monteiro@tecnico.ulisboa.pt
Claudine Chaouiya
chaouiya@igc.gulbenkian.pt

[†] *These authors have contributed equally to this work*

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: *06 April 2018*
Accepted: *02 August 2018*
Published: *07 September 2018*

Citation:
Mendes ND, Henriques R, Remy E, Carneiro J, Monteiro PT and Chaouiya C (2018) Estimating Attractor Reachability in Asynchronous Logical Models. Front. Physiol. 9:1161. doi: 10.3389/fphys.2018.01161

Logical models are well-suited to capture salient dynamical properties of regulatory networks. For networks controlling cell fate decisions, cell fates are associated with model attractors (stable states or cyclic attractors) whose identification and reachability properties are particularly relevant. While synchronous updates assume unlikely instantaneous or identical rates associated with component changes, the consideration of asynchronous updates is more realistic but, for large models, may hinder the analysis of the resulting non-deterministic concurrent dynamics. This complexity hampers the study of asymptotical behaviors, and most existing approaches suffer from efficiency bottlenecks, being generally unable to handle cyclical attractors and quantify attractor reachability. Here, we propose two algorithms providing probability estimates of attractor reachability in asynchronous dynamics. The first algorithm, named Firefront, exhaustively explores the state space from an initial state, and provides quasi-exact evaluations of the reachability probabilities of model attractors. The algorithm progresses in breadth, propagating the probabilities of each encountered state to its successors. Second, Avatar is an adapted Monte Carlo approach, better suited for models with large and intertwined transient and terminal cycles. Avatar iteratively explores the state space by randomly selecting trajectories and by using these random walks to estimate the likelihood of reaching an attractor. Unlike Monte Carlo simulations, Avatar is equipped to avoid getting trapped in transient cycles and to identify cyclic attractors. Firefront and Avatar are validated and compared to related methods, using as test cases logical models of synthetic and biological networks. Both algorithms are implemented as new functionalities of GINsim 3.0, a well-established software tool for logical modeling, providing executable GUI, Java API, and scripting facilities.

Keywords: regulatory network, logical modeling, discrete asynchronous dynamics, attractors, reachability

1. INTRODUCTION

Logical modeling has been widely used to study gene regulatory and signalling networks (see e.g., Glass and Siegelmann, 2010; Saadatpour and Albert, 2012; Abou-Jaoudé et al., 2016). Briefly, in a logical model, the evolution of the discretised level of each component depends on the current values of its regulators whose influences are dictated by logical functions. Here, we rely on the generalized framework initially introduced by Thomas and d'Ari (1990) and implemented in our software tool GINsim (Chaouiya et al., 2012; Naldi et al., 2018). Because precise knowledge of the

durations of underlying mechanisms is often lacking, one assumes that, when multiple components are called to change their levels, all update orders have to be considered. This corresponds to the asynchronous updating scheme (Thomas and d'Ari, 1990; Thomas, 1991). The dynamics of these models are classically represented by State Transition Graphs (STGs) where nodes embody the model states and edges represent the state transitions; each path in this graph accounts for a potential trajectory of the system. In contrast, synchronous updates, which amount to consider equal or negligible delays associated to component changes, define deterministic dynamics, easier to analyse but less realistic.

Model attractors (stable states or cyclic attractors) represent long term, stable equilibria. Cyclic attractors denote stable oscillations as observed in cell cycle or circadian rythms (see e.g., Fauré et al., 2006; Fauré and Thieffry, 2009; Chaves and Preto, 2013), whereas stable states are associated with cell lineages or other cellular responses to external cues or perturbations (see e.g., Sánchez et al., 2008; Calzone et al., 2010; Naldi et al., 2010; Collombet et al., 2017). Modeling molecular networks involved in cancer has been focusing on attractors and their reachability properties (see e.g., Huang et al., 2009; Flobak et al., 2015; Remy et al., 2015; Cho et al., 2016). Indeed, attractor likelihood may provide relevant predictions as attractors reflect cellular responses (e.g., healthy or not). For instance, to uncover patterns of genetic alterations in bladder tumors, Remy et al. (2015) considered an asynchronous logical model and checked how model perturbations modify the probabilities of reaching attractors related to proliferative phenotypes.

Not surprisingly, the number of states of logical models grows exponentially with the number of regulatory components. Moreover, due to the asynchronous updating scheme, the dynamics are non-deterministic; they possibly encompass alternative trajectories toward a given state as well as transient cycles. All this turns the identification and reachability analysis of model attractors into a difficult challenge. In this context, methods have been developed to find stable states—also referred as point attractors—and complex, oscillatory attractors (or, at least to circumscribe their location) (Naldi et al., 2007; Garg et al., 2008; Zañudo and Albert, 2013; Klarner et al., 2015). Here, we primarily aim at efficiently determining attractors reachable from specific initial condition(s) as well as estimating the reachability probability of each of those attractors in asynchronous dynamics.

An STG can be readily interpreted as the transition matrix of a finite Markov Chain. Generally, STGs encompass distinct attractors (or recurrent classes) and thus define absorbing chains (Grinstead et al., 1997). However, most existing results relate to recurrent (or irreducible) chains (Prum, 2012). Moreover, we aim at avoiding the construction of the whole dynamics (or the associated transition matrix); we thus rely on the logical rules as implicit descriptions of state transitions. Finally, we have here a specific interest on reachability properties.

Following a background section, we present two approaches to assess reachable attractors. First, the FIREFRONT algorithm is a quasi-exact method that starts from an initial state and simultaneously follows all (concurrent) trajectories while propagating state probabilities. This algorithm follows a principle similar to those employed for infinite Markov chains (Munsky and Khammash, 2006; Henzinger et al., 2009). To enable state space sampling and tackle models with large transient cyclic behaviors, we developed AVATAR, which is a Monte Carlo approach adapted to cope with strongly connected components. Both methods have been implemented as new functionalities of the software tool GINSIM (Naldi et al., 2018). They are applied to a range of models, illustrating their respective performances and specificities.

2. METHODS

In this section, we first briefly introduce the basics on Logical Regulatory Graphs (LRGs), their state transition graphs (STGs), attractors as well as absorbing Markov chains. We then present the algorithm FIREFRONT. The rest of the section focuses on AVATAR, an adaptation of the classical Monte Carlo simulation to cope with cyclical behaviors. It is worth noting that for small enough models it is possible to explicitly construct the STGs and identify reachable attractors, but it is not straightforward to evaluate their reachability probabilities.

2.1. Background

2.1.1. Basics on Logical Models and Their Dynamics

Definition 1. A Logical Regulatory Graph (LRG) is a pair (G, K), where:

- $G = \{g_i\}_{i=0,\dots n}$ is the set of regulatory components. Each $g_i \in G$ is associated to a variable v^i denoting its level, which takes values in $D_i = \{0, \dots M_i\} \subsetneq \mathbb{N}$; $v = (v^i)_{i=0,\dots n}$ is a state of the system, and $S = \prod_{i=0,\dots n} D_i$ denotes the state space.
- $(K_i)_{i=0,\dots n}$ denotes the *logical regulatory functions* (or logical rules); $K_i : S \to D_i$ is the function that specifies the evolution of g_i; $\forall v \in S$, $K_i(v)$ is the target value of g_i that depends on the state v.

The asynchronous dynamics of an LRG is represented by a graph as follows.

Definition 2. Given a logical regulatory graph (G, K), its asynchronous State Transition Graph (STG) is denoted (S, T), where:

- S is the state space,
- $T = \{(v, v') \in S^2 \mid v' \in \mathrm{Succ}(v)\}$, where for each state v, $\mathrm{Succ}(v) : S \to 2^S$ is the set of successor states w, satisfying the asynchronous property (one component is updated at a time):

$$\exists g_i \in G \text{ with } \begin{cases} K_i(v) \neq v^i \text{ and } w^i = v^i + \frac{K_i(v) - v^i}{|K_i(v) - v^i|}, \\ \forall g_j \in G \setminus \{g_i\}, \quad w^j = v^j. \end{cases}$$

Note that, from the STG defined above, one can consider the sub-graph reachable from a specific initial state v_0 or from a set of states $\{v_i\}_{i \in \{0,\dots m\}} \subseteq S$.

We further introduce some notation and classical notions.

Given an STG (S, T), we write $v \longrightarrow v'$ if and only if there exists a path between the states v and v'. In other words, there is a sequence of states of S such as: $v_0 = v, v_1, \dots v_{k-1}, v_k = v'$,

and for all $j \in \{1, \dots k\}$, $(v_{j-1}, v_j) \in T$. Furthermore, we denote $v \xrightarrow{k} v'$ such a path of length k.

A Strongly Connected Component (SCC) is a maximal set of states $A \subseteq S$ such that $\forall v, v' \in A$ with $v \neq v'$, $v \longrightarrow v'$. This is to say, there is a path between any two states in A, and this property cannot be preserved adding any other state to A.

Attractors of an LRG are defined as the *terminal* SCCs of its STG (i.e., there is no transitions leaving the SCC). If a terminal SCC is a single state we call it a *stable state*, otherwise it is a *complex attractor*.

2.1.2. Markov Chains and Absorption

The incidence matrix of an STG (S, T) naturally translates into an $|S| \times |S|$-transition matrix Π, which is a stochastic matrix (for all $v \in S$, $\sum_{u \in S} \Pi(v, u) = 1$):

$$\begin{aligned} \forall v, v' \in S \quad & \Pi(v, v') > 0 \Leftrightarrow (v, v') \in T, \\ \forall v \in S \quad & \Pi(v, v) = 1 \Leftrightarrow Succ(v) = \emptyset, \\ & \Pi(v, v) = 0 \text{ otherwise.} \end{aligned}$$

We assume that probabilities of concurrent transitions are uniformly distributed: $\forall v \in S, \forall v' \in Succ(v)$, $\Pi(v, v') = 1/|Succ(v)|$. Extension to other distributions would be rather straightforward.

A Markov chain (μ_0, Π) is defined by the finite set S, the transition matrix Π, and the initial law μ_0 (that depends on the selection – or not – of an initial condition). We want to define the chain stopped when it reaches an attractor. For that, we consider the quotient graph of (S, T) with respect to the equivalence relation: $u \sim v \Leftrightarrow u \longrightarrow v$ and $v \longrightarrow u$. In this quotient graph, each node gathers a set of states and corresponds to a class of the Markov chain. The absorbing nodes of the quotient graph (i.e., nodes with no output arcs) form the absorbing classes of the chain (μ_0, Π), all the other classes being transient. Note that the number of absorbing classes is the number of attractors of the corresponding STG. Let θ be this number and $a_1, \dots a_\theta$ the absorbing classes.

Now, let us stop the chain (μ_0, Π) when it reaches an absorbing class: we thus define the Markov chain X on the set $\tilde{S} = \mathcal{T} \cup \mathcal{A}$, where $\mathcal{T} \subset S$ is the set of all the transient states, and $\mathcal{A} = \{\{a_i\}, i = 1, \dots \theta\}$ (each element a_i being an absorbing class). The transition matrix π of X is:

$$\begin{aligned} & \pi(u, a_i) = \sum_{v \in a_i} \Pi(u, v) \quad \forall u \in \mathcal{T}, \forall a_i \in \mathcal{A}, \\ & \pi(a_i, u) = 0 \qquad \forall u \in \mathcal{T}, \forall a_i \in \mathcal{A}, \\ & \pi(a_i, a_i) = 1 \qquad \forall a_i \in \mathcal{A}, \\ & \pi(a_i, a_j) = 0 \qquad \forall a_i \in \mathcal{A}, \forall a_j \in \mathcal{A}, i \neq j, \\ & \pi(u, v) = \Pi(u, v) \qquad \forall u, v \in \mathcal{T}. \end{aligned}$$

Reordering the states by considering first the transient ones, (i.e., those belonging to $\mathcal{T}$) and then the absorbing classes (i.e., the elements of $\mathcal{A}$), the transition matrix π is under its canonical form:

$$\pi = \begin{pmatrix} Q & L \\ 0 & I \end{pmatrix},$$

where $Q(u, v) = \pi(u, v)$ for $u, v \in \mathcal{T}$, $L(u, a) = \pi(u, a)$ for $u \in \mathcal{T}$ and $a \in \mathcal{A}$, 0 is the null matrix (no transition from an absorbing class to a transient state), and I the identity matrix. One can easily verify that:

$$\pi^k = \begin{pmatrix} Q^k & (\sum_{j=0}^{k-1} Q^j) L \\ 0 & I \end{pmatrix},$$

$\pi^k(u, v)$ denotes the probability that, started in state u, the chain is in state v after k steps: $\pi^k(u, v) = \mathbb{P}_u(X_k = v) \triangleq \mathbb{P}(X_k = v \,|\, X_0 = u)$. Proofs of the next, well-known results can be found in [e.g., (Grinstead et al., 1997), chap. 11].

- Q^k tends to 0 when k tends to infinity, and

$$\lim_{n \to +\infty} \sum_{k=0}^{n} Q^k = (I - Q)^{-1}. \tag{1}$$

- The hitting time of $\mathcal{A}$ is almost-surely finite.
- From any $u \in \mathcal{T}$, the probability of X being absorbed in $a \in \mathcal{A}$ is $\mathbb{P}_u(X_\infty = a) = (Id - Q)^{-1} L(u, a)$.

By an abuse of terminology, we will refer to $\mathbb{P}_u(X_\infty = a)$ as the probability to reach the attractor a from the initial state u.

2.2. Firefront

FIREFRONT is our first method to identify attractors and assess their reachability probabilities. Although simple, it is effective for restricted types of dynamics as demonstrated in section 4. Briefly, the algorithm progresses in breadth from an initial state v_0, which is first assigned probability 1. It distributes and propagates the probability of each visited state to its successors, according to the transition matrix Π.

At any step k, the set of states being expanded and carrying a fraction of the original probability is called *firefront* as it corresponds to the front line of the breadth-first exploration of the STG: $F_k = \{v \in S, \exists v_0 \xrightarrow{k} v\}$. Basically this procedure, called expansion, calculates at each iteration k and for each state v the probability of the Markov chain X to be in v after k steps from state v_0: $\mathbb{P}_{v_0}(X_k = v) = \pi^k(v_0, v)$. Clearly, by the definition of the set F_k, $\mathbb{P}_{v_0}(X_k \in F_k) = 1$; the firefront will ultimately contain only states that are stable states or members of complex attractors. In what follows, we will simply denote the firefront set F, omitting the index k. Actually, attractors are not kept in F, they are instead stored in another set A (see below), hence F becomes ultimately empty.

In practice, to tackle efficiency bottlenecks avoiding the exploration of unlikely trajectories, we introduce a set of *neglected states* N. Furthermore, to ensure that the algorithm terminates whenever the reachable attractors are all stable states, we consider the set of *attractors* A. In the course of the exploration the firefront F is reduced as explained below:

- if the probability associated with a state $v \in F$ drops below a certain value α, then v is moved from F to N (set of neglected states). As a consequence, the immediate successors of v will

not be explored at this time. If a state $v \in N$ is visited again as being the successor of a state in F, its probability is properly updated (we will say that it accumulates more probability), and if this probability exceeds α, then v is moved from N back to F (see **Figure 1**, step 7);
- if a state in F has no successors, it is moved to A (set of stable states); if it is already in A, its probability increases according to this new trajectory.

At each step, the sum of the probabilities of the states in F, N, and A is 1.

Algorithm 1 FIREFRONT

Input: α, β, v_0 // min prob. to stay in F, total prob. in F under which the procedure halts, initial state
Output: A // set of reachable attractors with their probabilities

$F \leftarrow \{v_0\}$ $N \leftarrow \emptyset$ $A \leftarrow \emptyset$
while total probability in $F > \beta$ **do**
  $F' \leftarrow \emptyset$
  while $F \neq \emptyset$ **do**
    $v \leftarrow$ select and remove element of F
    if Succ(v) $= \emptyset$ **then**
      v is added to A as a stable state
    else
      for all $v' \in$ Succ(v) **do**
        $p \leftarrow$ divide $p(v)$ by $|\text{Succ}(v)|$
        if v' is in F', N or A **then**
          Add p to the probability of v'
        else
          Set the probability of v' to p
        end if
        if probability of $v' \geq \alpha$ **then**
          Add v' to F' if it is not in A
          Remove v' from N if it is there
        else
          Add v' to N
        end if
      end for
    end if
  end while
  $F \leftarrow F'$
  if isOscillating(F) **then**
    Extract complex attractors: move their states from F and N into A
  end if
end while

Unlike forest fires, which do not revisit burnt areas, the algorithm will, in general, revisit the same state in the presence of a cycle. This invalidates our colorful metaphor unless imagining uncannily rapid forest regeneration. The presence of cycles thus poses some difficulties because the algorithm would never terminate. To address this issue, FIREFRONT detects periodicities of the ensemble of states entering and exiting F (i.e., states with a sustained oscillating probability); three sequential occurrences of exactly the same set F are assumed to be sufficient evidence that the simulation is locked within a complex attractor. In this situation, all the states found in F between the second and third occurrences are used to compose the complex attractor. To do so efficiently, FIREFRONT uses a reversible hash-function. This heuristic thus enables the identification of complex attractors from oscillating behaviors throughout expansions. Nevertheless, since FIREFRONT progression can still become locked in large and complex cycles for a lengthy number of expansions, the user may specify a maximum depth (number of expansions) to guarantee its termination in useful time.

When available, the algorithm can be provided with a description of the complex attractors, equipping FIREFRONT with a function called *oracle* that indicates whether a state belongs to a listed complex attractor. In this case, FIREFRONT halts the exploration whenever it reaches a state recognized by the oracle, and treats all members of the corresponding attractor as a single element of A collectively accumulating incoming probabilities.

FIREFRONT terminates when: 1) the total probability in F drops to zero or below some predefined threshold β, or 2) the predefined maximum depth is reached. Given the initial state v_0, the probability associated to each attractor $a \in A$ is a lower bound of $\mathbb{P}_{v_0}(X_\infty = a)$. An upper bound is obtained by adding to this value β and the sum of probabilities accumulated in N. An outline of FIREFRONT is presented in Algorithm 1, and **Figure 1** provides an illustration on a toy example.

2.3. Avatar

AVATAR is proposed as an alternate algorithm to identify model attractors and quantify their reachability, considering specific initial state(s) or the whole state space. AVATAR is an adaptation of the classical Monte Carlo simulations that aims at efficiently coping with (transient and terminal) SCCs.

2.3.1. The Algorithm

When exhaustive enumeration is not feasible, Monte Carlo simulation is classically used to estimate the likelihood of an outcome. Concerning attractor reachability in logical models, this means following random paths along the asynchronous dynamics (the STG). Each simulation halts when either a stable state (with no successor) or the maximal depth are reached. Performing a large number of simulations allows estimating reachability probabilities of stable states. The simulation does not record past states, and thus memory requirements are minimal. However, a major drawback is that cycles are not detected. Consequently, without restricting the number of steps, the simulation does not terminate when a trajectory enters a terminal SCC. Moreover, in the presence of a transient cycle, it may re-visit the same states an unbounded number of times before exiting. That is why we propose an appropriate modification of this approach.

AVATAR is outlined in Algorithm 2 (further description of AVATAR and its ancillary procedures is provided in the **Supplementary Material S1**). It avoids repeatedly visiting states by detecting that a previously visited state is reached, indicating the presence of a cycle in the dynamics. Having detected a cycle, the algorithm modifies the STG in order to dismantle the cycle, linking its states to its exiting states (i.e., targets

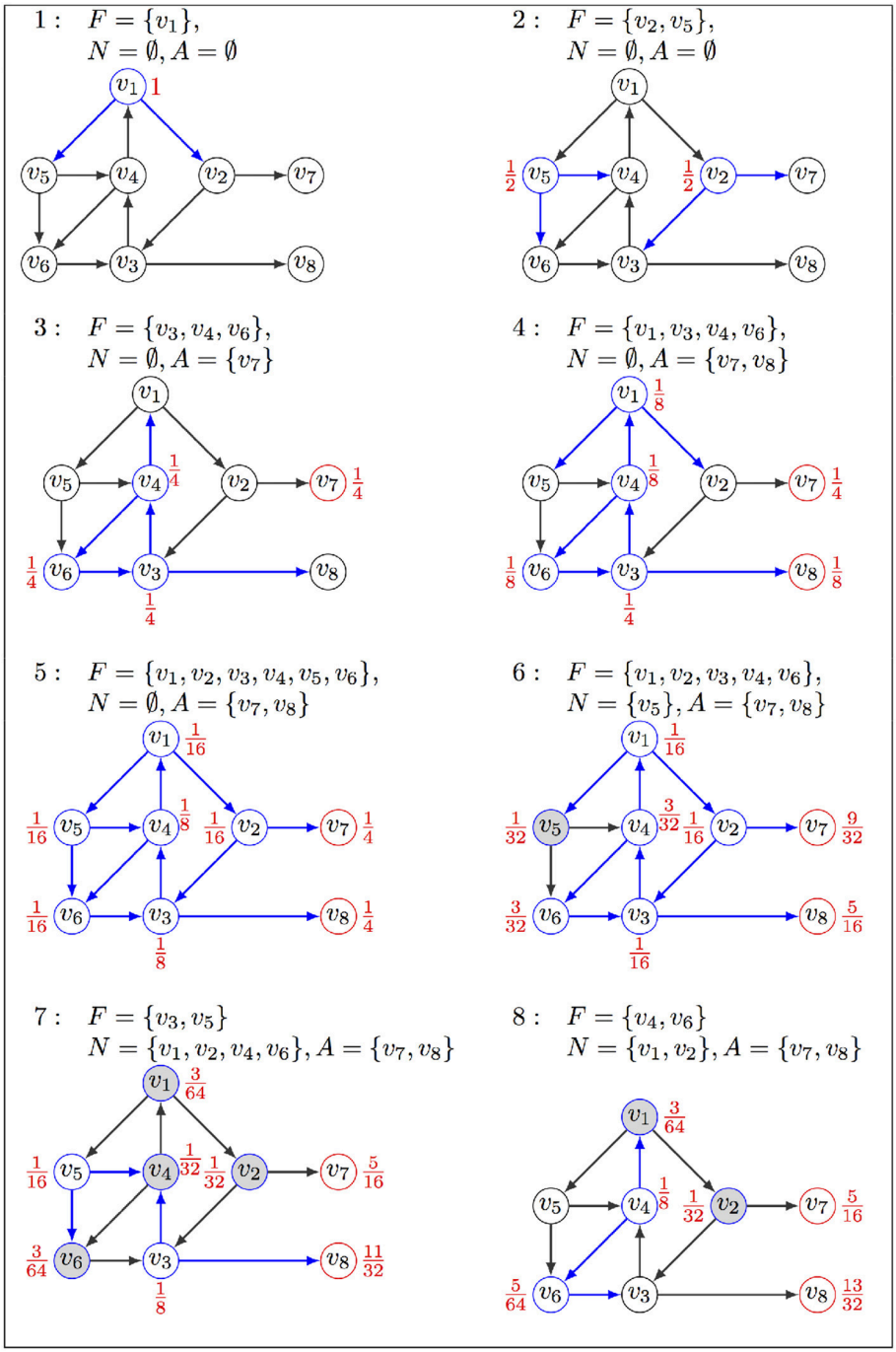


FIGURE 1 | Illustration of FIREFRONT operation, with $\alpha = \frac{1}{16}$: (1) The exploration starts from initial state v_1 in F associated with probability 1, sets A and N are empty; (2) successors replace v_1 in F, associated with their probabilities; (3–4) states in F are replaced by their successors, but the stable state v_7 goes in A; (4) v_3, v_4, v_6 stay in F with updated probabilities; (5) probability of v_8 in A increases as it is visited again; (6) v_5 goes to N as its probability is lower than α; (7) v_5 is removed from N and put back in F as its probability increased when visited again from v_1. Transitions explored in the current iteration are in blue, their sources being labeled with their probabilities. Red nodes are in A, and gray nodes are in N. The exploration will halt when F is empty or the maximum number of iterations is reached.

of transitions leaving the cycle). It is important, however, to associate these new transitions with appropriate probabilities; the probability of a transition from any cycle state to a given exit must match the corresponding asymptotic probability, considering the infinitely many possible trajectories. The STG is thus rewired so as to replace all the transitions between the cycle states by

transitions from each cycle state toward each cycle exit (see **Figure 2**). Each rewiring creates a new so-called *incarnation* of the dynamics. Such an incarnation—Sanskrit name of our algorithm—is a graph with the same states as the original STG, but with different transition probabilities. This rewiring relies on theoretical foundations that are presented in section 2.3.2. Upon rewiring, the simulation proceeds from the current state.

Because it is generally more efficient to rewire a large transient than to iteratively rewire portions of it, upon encountering a cycle, AVATAR performs an extension step controlled by a parameter τ that is a modified Tarjan's algorithm for SCC identification (Tarjan, 1972)—trajectories exploration is performed up to a depth of τ away from states of the original cycle. The subsequent rewiring is then performed over the (potentially) extended cycle. In the course of a single simulation, the value of τ is doubled within each attempt to enlarge a cycle in order to speed up the identification of large transients.

When a newly visited state v has no successor, it is a stable state. But if v was part of a cycle in a previous incarnation, v belongs to a complex attractor, which is computed as the equivalence class containing all the cycles that included v in past incarnations.

As for FIREFRONT, the algorithm can be complemented with the previous knowledge of the attractors (oracles). This obviously improves AVATAR's performance. Moreover, AVATAR not only evaluates the probability of the attractors being reached from an initial condition, it can also be used to assess the probability distribution of the attractors for the whole state space (i.e., considering all possible initial states). AVATAR is also able to use the knowledge regarding the identified transient SCCs within one iteration to alleviate the cost of identifying and possibly rewiring large cycles in upcoming iterations, thus boosting the overall efficiency of the simulation. The knowledge regarding the sizes of the transient SCCs and average depths of the found attractors can provide valuable insights into the model dynamics.

Algorithm 2 AVATAR (single simulation)

Input: v_0
Output: A // attractor set
$t \leftarrow 0$ // incarnation counter
$v \leftarrow v_0$ // initial state
while v has successors **do**
  $v' \leftarrow$ successor of v chosen with probability $\pi(v,v') = 1/|Succ(v)|$
  if v' was already visited in incarnation t **then**
    $C^t \leftarrow$ set of all states visited since the discovery of v'
    Extend cycle C^t
    $B \leftarrow$ set of exits //successors of states in C^t that are not in C^t
    if $B = \emptyset$ // C^t has no exits **then**
      $A \leftarrow C^*$ where $\forall w \in C^*$, if $\exists k$ s.t. $w \in C^k$ then $C^k \subseteq C^*$
    else
      // Rewire the graph
      $q \leftarrow \left[\pi(v,w)\right]_{v,w\in C}$
      $r \leftarrow \left[\pi(v,w)\right]_{v\in C, w\in X}$
      $r^1 \leftarrow \left(\mathrm{Id}_{|C|\times|C|} - q\right)^{-1} r$
      for all $v \in C$ **do**
        for all $w \in C$ **do**
          $\pi(v,w) \leftarrow 0$
        end for
        for all $w \in B$ **do**
          $\pi(v,w) \leftarrow r^1_{v,w}$
        end for
      end for
    end if
    $t \leftarrow t+1$
  end if
  $v \leftarrow v'$
  if v has no successors **then**
    $A \leftarrow v$ //stable state
  end if
end while

2.3.2. Theoretical Foundations of Avatar Rewiring

The rewiring performed by AVATAR to force the simulation exiting a cycle modifies the probabilities associated to transitions. This is properly done so as to ensure a correct evaluation of the reachability probabilities performing a (large) number of random walks over our Markov chain X. This procedure amounts to modify the chain. It is formalized below and illustrated in **Figure 2**.

Suppose that $X_t = c_1$, and $X_{t+k} = c_1$ for t and k two positive integers. The walk has thus traveled along the cycle $C = (c_1, c_2, \ldots c_k)$ (with $c_i \in S$ and $(c_i, c_{i+1}) \in T$, $\forall i = 1, \ldots k$). Note that this cycle may contain "direct shortcuts": $(c_i, c_j) \in T, j \neq i+1 \pmod k$. We denote by B the set of states directly reachable from C: $B = \{v \in S \setminus C, (c_i, v) \in T, c_i \in C\}$. Let q be the $k \times k$ sub-matrix of π, for states $c_1, \ldots c_k$, and r the $k \times |B|$ sub-matrix of π, defining transitions from C to B. To force the walk leaving the cycle (rather than being trapped there for a long time), the transition matrix is modified as follows:

- remove the transitions between the states of C; the sub-matrix q is replaced by $q^1 = 0$, the null matrix;
- add an arc from each state of C to each state of B; the sub-matrix r is replaced by $r^1 \triangleq \sum_{j=0}^{\infty} q^j r$. By Equation (1), section 2.1.2, $\forall c_i \in C$, $\forall v \in B$, $r^1(c_i, v) = \left[(Id - q)^{-1} r\right](c_i, v)$.

Y denotes this new chain. Property 1 asserts that, starting from any transient state u, X, and Y have the same asymptotical behaviors.

Property 1. $\forall u \in \mathcal{T}, \forall a \in \mathcal{A}, \mathbb{P}_u(Y_\infty = a) = \mathbb{P}_u(X_\infty = a)$.

Proof: Transition matrices of X and Y are the same except around the states of the cycle C; they behave differently only when traveling along C: from c_i, entry state of C, X runs along C for l steps ($l \geq 0$), leaving C through a state $v \in B$ with probability $q^l r(c_i, v)$, whereas Y would go directly from c_i to v, with probability $r^1(c_i, v)$. Hence, for all $u \in \mathcal{T}$, $a \in \mathcal{A}$ and $j \geq 0$,

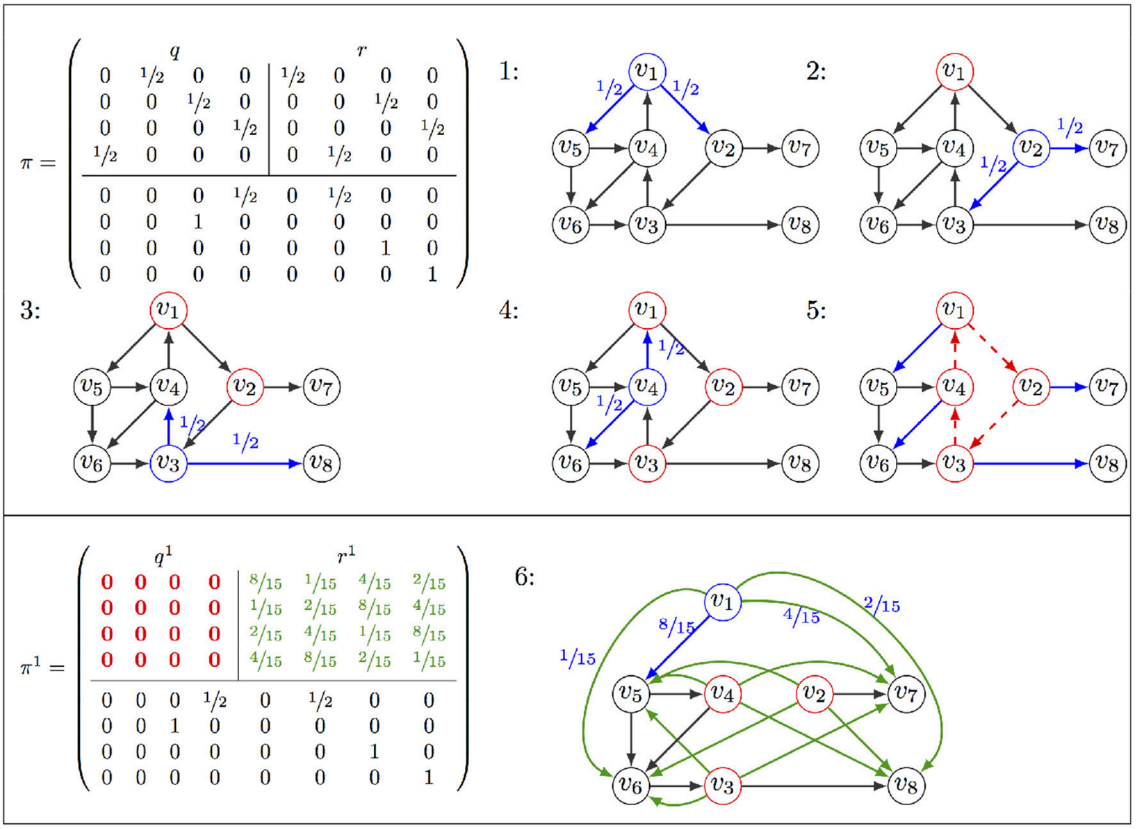


FIGURE 2 | Illustration of AVATAR operation: The transition matrix π is partitioned into the sub-matrices q for transitions between states $v_1, \ldots v_4$ of the cycle to be discovered **(Top Left)**, and r for transitions leaving the cycle **(Top Right)**. Exploration starts at v_1 (denoted in blue as well as its leaving transitions with their probabilities), v_2 is selected for the second iteration, and v_1 is indicated as being already visited in red. Exploration proceeds until revisiting v_1 at the 5^{th} step. Having identified a cycle, the rewiring procedure is launched, removing transitions of the cycle (dotted red) and adding transitions toward exits (green). Probabilities are computed, resulting in a new matrix π^1, with $q^1_{ij} = 0$ and $r^1_{ij} = ((Id - q)^{-1} r)_{ij}, i = 1, \ldots 4$. From v_1, an exit of the cycle is chosen according to these probabilities (step 6).

we have $\mathbb{P}_u(Y_j = a) \geq \mathbb{P}_u(X_j = a)$ and thus,

$$\sum_{j=1}^{k} \mathbb{P}_u(Y_j = a) \geq \sum_{j=1}^{k} \mathbb{P}_u(X_j = a),$$
$$\mathbb{P}_u(Y_\infty = a) \geq \mathbb{P}_u(X_\infty = a),$$
$$1 = \sum_{a \in \mathcal{A}} \mathbb{P}_u(Y_\infty = a) \geq \sum_{a \in \mathcal{A}} \mathbb{P}_u(X_\infty = a) = 1.$$

All the terms being positive, the Property is proved. Therefore, the rewiring does not asymptotically affect the output of the simulation. □

Despite the inherent simplicity and time efficiency of the rewiring step, its dependency on matrix inversions can lead to a memory bottleneck for very large cycles. As such, the current implementation of AVATAR uses a ceiling size for a cycle to be rewired. When AVATAR finds a cycle, it still attempts to extend it as far as possible. If the extended cycle has some exits, it needs to be rewired. However, if the extended cycle has more states than the specified ceiling, only a sub-cycle (with as much states as allowed) of the detected cycle is rewired. Furthermore, the user can also choose an approximate strategy for rewiring that still guarantees the selection of exit states when entering a cycle without the need to perform an exact estimation of their likelihood. This is done by assigning uniform probabilities from the states of a cycle to its exits. Although this strategy is not prone to memory bottlenecks, its approximate nature can lead to biases on the computed reachability probabilities.

3. IMPLEMENTATION

Both FIREFRONT and AVATAR are implemented in the context of GINSIM, which supports the definition and analysis of logical models (Chaouiya et al., 2012; Naldi et al., 2018). **Figure 3** provides a snapshot of the desktop GUI, showing the selection of the algorithm, specification of model modifications (perturbation or reduction), initial conditions, and algorithm parameters. MONTECARLO simulations are also available, as well as a modified version of AVATAR with the approximate strategy described above. User documentation of Firefront and Avatar is provided in the **Supplementary Material S2**.

The implementations of FIREFRONT and AVATAR rely on adequate data structures—states are easily indexable through

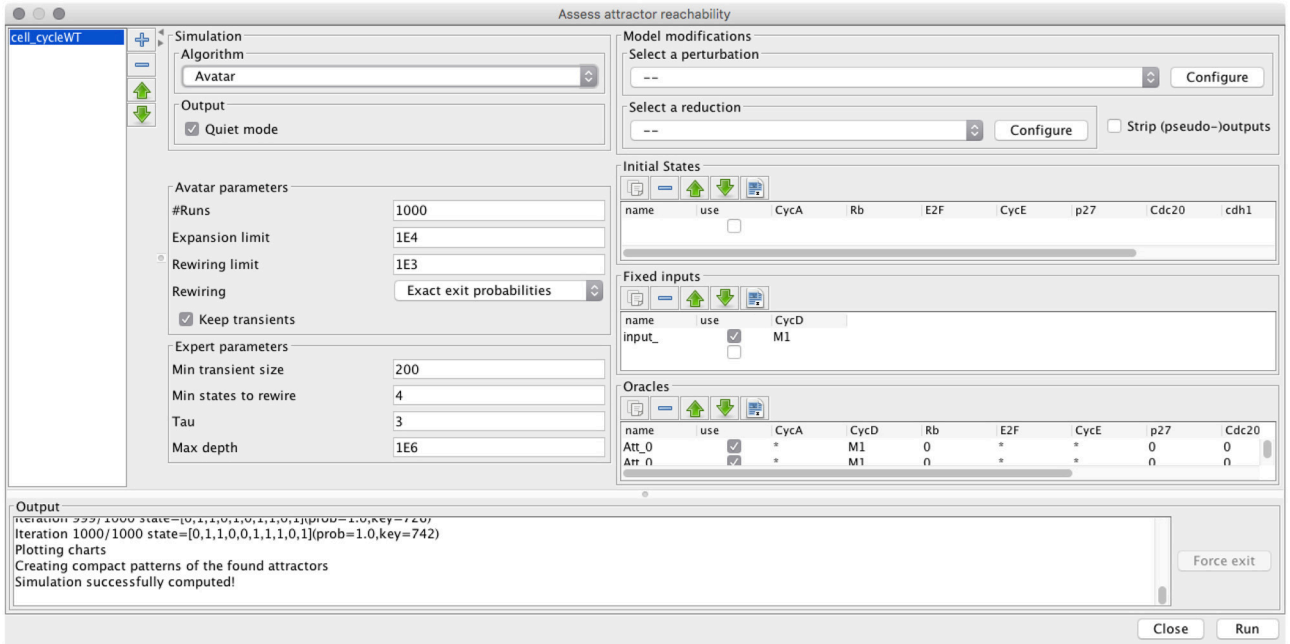


FIGURE 3 | GUI for the assessment of attractor reachability within GINSIM.

meaningful and compact hash keys, and sets of states are implemented as a map of states for highly efficient indexations, additions and removals. Our implementation of FIREFRONT halts the STG exploration after a predefined number of expansions (10^3 by default). AVATAR implementation includes a heuristic optimization controlled by optional parameters whose default values were found to be appropriate for the tested models. This optimization considers tradeoffs between costly rewirings and simulations freely proceeding along cycles, as well as between memory cost of keeping state transitions after rewiring and not profiting from rewirings in previous simulations. AVATAR further supports sampling over (portions of) the state space. In this case, iterations within a simulation start from states randomly selected over the unconstrained model components.

Both algorithms provide textual and visual displays of the results: attractors and their reachability probabilities, maximal size of encountered transient SCCs, and plots of the evolution of the set contents for FIREFRONT and of the probability estimates for AVATAR (see section 4).

4. RESULTS

To validate the proposed algorithms, we considered a number of case studies including randomly generated, synthetic and published biological models. All are briefly described below. We analyzed how FIREFRONT and AVATAR perform on these case studies and compared, when possible, to outcomes produced by BOOLNET (Müssel et al., 2010) and MONTECARLO simulations. BOOLNET is an R package not only able to generate random Boolean models, but also to identify attractors and to perform Markov chain simulations. We further compared AVATAR with MABOSS, a C++ software implementing a Monte Carlo kinetic algorithm to produce time trajectories of Boolean models (Stoll et al., 2012), and with the probabilistic model checker PRISM (Kwiatkowska et al., 2011, 2017). The experiments were run using an Intel(R) i7-7500U CPU @ 2.7GHz and 8GB of RAM.

4.1. Case Studies Description

Two sets of synthetic models were generated. First, we used BOOLNET (Müssel et al., 2010) to define random models with 10 to 15 components, each with 2 regulators and logical rules randomly selected (uniform distribution)[1]. From the resulting set of random models, three models were selected for exhibiting multi-stability (**Table 1**). Additionally, we constructed a "synthetic" model exhibiting a large complex attractor and a few transient cycles. To further challenge our algorithms, we modified this last model, adding one component in such a way that the complex attractor turned into a transient cycle with very few transitions leaving toward a stable state (see synthetic models 1 and 2 in **Table 1**).

Our case studies also include published biological models. First, a Boolean model of the mammalian cell cycle control (Fauré et al., 2006), which has 10 components and exhibits one stable state (quiescent state) and one complex attractor (cell cycle progression). These attractors arise in (two) disconnected regions of the state space, controlled by the value of the sole input

[1]This process is automated in BOOLNETR2GINSIM, a small program available at https://github.com/ptgm/BoolNetR2GINsim that accepts user-defined parameters, calls BOOLNET and writes the resulting model to a GINML file (the GINSIM format).

TABLE 1 | Characteristics of the models used as case studies to challenge Firefront and Avatar: type of variables (Boolean vs. multi-valued), number of input components (these remain constant) and internal components, number and type of attractors with number of states in the case of complex attractors, size of the state space with the total number of model states.

Model name	Boolean	# Components		# Attractors		# States
	(Y/N)	Input	Internal	Stable states	Complex attractors (*size*)	
Random 1	Y	0	10	1	1 (4)	1 024
Random 2	Y	0	10	1	1 (4)	1 024
Random 3	Y	0	15	1	1 (4)	32 768
Synthetic 1	Y	0	15	1	1 (8192)	32 768
Synthetic 2	Y	0	16	2	0	65 536
Mammalian Cell Cycle	Y	1	9	1	1 (112)	1 024
Segment Polarity (*sp1*, 1-cell)	N	2	12	3	0	186 624
Segment Polarity (*sp2*, 2-cells)	N	0	24	3	0	$\approx 9.7 \times 10^8$
Segment Polarity (*sp4*, 4-cells)	N	0	48	15	0	$\approx 9.4 \times 10^{17}$
Bladder model	N	4	26	20	5 (16,16,32,512,184320)	$\approx 8.5 \times 10^9$

component (CycD, which stands for the presence of growth factors).

Second, Sanchez et al.'s multi-valued model of the segment polarity module—involved in early segmentation of the *Drosophila* embryo—defines an intra-cellular regulatory network. Instances of this network are connected through inter-cellular signaling (Sánchez et al., 2008). Here, we consider three cases: 1) the intra-cellular network (one cell), 2) the composition of two instances (i.e., two adjacent cells), and 3) the composition of four instances. Initial conditions are specified by the action of the pair-rule module (Wg-expressing cell for the single cell model) that operates earlier in development (see Sánchez et al., 2008 for details).

Third, we consider the interaction network of genes frequently altered in bladder cancer as proposed in Remy et al. (2015). This model includes 4 input components leading to different responses (EGFR, FGFR3 stimuli, Growth inhibitor, DNA damage), 23 internal components and 3 output components representing cellular responses or phenotypes (Proliferation, Apoptosis, Growth Arrest). Depending on the input values, the model displays multistability or not, with a combination of stable states and complex attractors. This case study further demonstrates the capacity of AVATAR in assessing large complex attractors, quantifying attractor reachability, and revealing transient dynamics.

Finally, using a model of T helper cells differentiation (Naldi et al., 2010) and a model of cell fate decision in response to death receptor engagement (Calzone et al., 2010), we provide additional illustrations in the **Supplementary Materials S4**, **S5**.

Supplementary Material S6 provides an archive containing all the models in the GINsim format (zginml).

4.2. Firefront and Avatar in Action

Results are summarized in **Table 2**. Generally, FIREFRONT and AVATAR show efficiency gains against alternatives and are further able to surpass the drawbacks of BOOLNET (applicable to Boolean models only) and MONTECARLO (unable to identify transient and terminal cycles).

Considering **random models 1 to 3**, FIREFRONT and AVATAR are able to efficiently find the stable states and complex attractors of these models and to estimate their reachability probabilities. BOOLNET is slower for these random models. MONTECARLO is not only less efficient but is also unable to detect the complex attractors. For instance, in random model 2, less than 8% of the simulations succeeded.

For **synthetic model 1**, FIREFRONT takes over a minute to distribute the probability out of the large transient cycles. For **synthetic model 2**, FIREFRONT could not distribute more than 5% of the probability out of the transient SCC (purposely constructed with 8 196 states and a dozen exits). The presence of multiple large transient SCCs causes FIREFRONT to accumulate a large number of states in F, leading to some time overhead and difficulty to distribute the probabilities. States of transient SCCs are revisited until the probabilities of their incoming transitions drop below α, which can take long. As such, the computational performance of FIREFRONT is greatly influenced by the structure of the STG (e.g., state outdegrees or sizes of transient SCCs). The **Supplementary Material S3** provides illustrations of the structures of the dynamics. In contrast, AVATAR is able to adequately identify and exit transient SCCs. For this reason, AVATAR was able to escape the transient SCC planted in synthetic model 2 thanks to its rewiring procedure, and could identify and quantify the attractors for both synthetic models. BOOLNET completed synthetic models 1 and 2, after 7 and 5 days, respectively, which highlights the need for the proposed methods to face efficiency bottlenecks for models with large and complex SCCs.

Starting in the region of the state space where the **mammalian cell cycle model** has a (unique) complex attractor (i.e., with the presence of CycD), AVATAR, FIREFRONT, and BOOLNET could assess its reachability from the quiescent state; when sampling the state space, both AVATAR and BOOLNET could correctly quantify the reachability of the two attractors (FIREFRONT was not applicable as it requires a starting initial state). Expectedly, MONTECARLO could not retrieve the complex attractor, being unable to exit it in all runs.

With regards to the **segment polarity model**, FIREFRONT was efficient for all cases (single, two and four cells), although

TABLE 2 | Summary of the results for FIREFRONT, AVATAR, BOOLNET, and MONTECARLO.

Name (initial state)		FIREFRONT					AVATAR				BOOLNET			MONTECARLO			
	# Reach	Time	Attract.	Prob. bounds	Residual	Expansions	Time	Attract.	Prob.	Largest transient	Time	Attract.	Prob.	Time	Attract.	Prob. (bounds)	Support (%simulations)
random1 (0000000000)	1024	1s	SS1 CA1	[0.674,0.678] [0.322,0.326]	4.2E-3	54	6 s	SS1 CA1	0.672 0.328	880	19 s	SS1 CA1	0.67 0.33	9 s	SS1	[0.91,1.00]	0.91
random2 (0100011100)	88	1s	SS1 CA1	[0.25,0.25] [0.75,0.75]	1E-4	40	4 s	SS1 CA1	0.256 0.744	4	19 s	SS1 CA1	0.25 0.75	14 s	SS1	[0.77,1.00]	0.77
random3 (100000000000000)	1408	1s	SS1 CA1	[0.21,0.21] [0.79,0.79]	3.4E-3	37	5 s	SS1 CA1	0.205 0.795	168	20 s	SS1 CA1	0.2 0.8	69 s	SS1	[0.08,1.00]	0.08
synthetic1 (000001100110111)	28320	93s	SS1	[0.51,1.00]	0.49	10000	12 s	SS1 CA1	0.586 0.414	1 024	8 days	SS1 CA1	0.6 0.4	72 s	SS1	[0.19,1.00]	0.19
synthetic2 (0000001000000100)	16224	86s	SS1 SS2	[0.01,0.96] [0.05,1.00]	0.95	10000	421 s	SS1 SS2	0.92 0.08	8 192	5 days	SS1 SS2	0.92 0.08	89 s	SS1	[0.006,1.00]	0.006
mmc quiescent & CycD=1	148	1s	CA1	[1.00,1.00]	7.8E-4	28	3s	CA1	1.00	4	195s	CA1	1.00	61s	–	–	0.0
mmc (sampling)	1024	NA due to sampling					2s	CA1 SS1	0.506 0.494	416	110s	CA1 SS1	0.5 0.5	65s	SS1	[0.51,1.00]	0.51
sp1 (Wg-exp.)	330	0.2s	SS1 SS2	[0.84,0.84] [0.16,0.16]	8.4E-4	42	3s	SS1 SS2	0.837 0.162	76	NA (multivalued)			1s	SS1 SS2	0.81 0.19	1.0
sp2 (pair rule)	3.2434E8	3s	SS1 SS2	[0.63,0.90] [0.10,0.37]	0.27	263	153s	SS1 SS2 SS3	0.8921 0.1078 1E-4	1 844 562	NA (multivalued)			1s	SS1 SS2 SS3	0.78 0.21 0.01	1.0
sp4 (pair rule)	unknown	25s	SS1 SS2 SS3 SS4	[0.11,0.97] [0.02,0.88] [0.01,0.87] [0.00,0.86]	0.86	333	3074s	SS1 SS2 SS3 SS4 SS5 SS6–9	0.8645 0.0628 0.0571 0.0133 0.0016 <1E-3	1 841 692	NA (multivalued)		5s	SS1 SS2 SS3 SS4 SS5 SS6–9	0.89 0.064 0.028 0.012 1E-3 <1E-3	1.0

Parameters: for Firefront, $\alpha = 10^{-5}$, $\beta = 10^{-5}$ *and maximum of expansions of* 10^4*; for* Avatar *and* MonteCarlo, *number of runs of* 10^4 *and maximum depth of* 10^4*; for* Avatar *rewiring limit of* 10^4, *expansion limit of* 10^3, *growth factor* $\tau = 3$ *and a minimum of* 200 *and* 4 *states to respectively save transients and apply rewiring. For* Firefront, *residual probabilities are the sums of those found in the firefront F and neglected N sets. Stable states are denoted SS and complex attractors CA.*

its ability to distribute all the probability decreases with the increase of model size. Since it did not reach the allowed maximum number of iterations, its stopping condition was that the total probability in F dropped below β, with all the residual probability in the neglected set, which in the end contained approximately 140, 52 000, and 210 000 states for the models of single, two and four cells, respectively. This would suggest that α was not small enough with respect to the number of concurrent trajectories toward the attractors (see **Supplementary Material S4** for illustration). Although AVATAR's performance is constrained by the need to assess the complex structure of the two and four cells' models (for instance the largest encountered transient SCC for *sp2* has over a million states), it is adequately able to find the attractors, even those with a low reachability probability. Given the fact that the attractors of these models are stable states, MONTECARLO was able to retrieve them, in particular those attractors reachable without the need to visit large transient SCCs.

Figure 4 complements these results by showing, for two of our case studies: with FIREFRONT, the evolution of the cardinals of the sets F, N, and A (and their corresponding probabilities), and with AVATAR, the convergence of the estimated reachability probabilities of the attractors.

The application of AVATAR over the **bladder tumourigenesis model**—with results illustrated in **Table 3**—enabled the quantification of attractor reachability over the whole state space, for 8 combinations of input values. Stable states were gathered in 3 classes, corresponding to the cell phenotypes Proliferation, Apoptosis and Growth Arrest, which are indicated by the values of the 3 output components of the model. The model displays several complex attractors. The reachability quantification of the attractors is relevant in the cases of multi-stability, i.e., when several attractors arise for the same input condition (compare with **Table S2** in Remy et al., 2015). AVATAR discloses structural properties of the model dynamics such as the sizes of encountered transient SCCs and mean depths of the attractors (not shown).

We also performed the analysis of model perturbations to illustrate the biological relevance of assessing attractor probabilities. To this end, we considered the case of activating mutations of fibroblast growth factor receptor 3 (FGFR3) and of the oncogene PI3K, one of the co-occurrent genetic perturbations observed in bladder tumors (see Remy et al., 2015). **Figure 5** illustrates how probabilities of the attractors are modified under those perturbations. It supports the conclusions drawn in Remy et al. (2015): mutating FGFR3 in PIK3-mutated tumors seems to be advantageous (to increase the probability of Proliferation); a third mutation is required for uncontrolled proliferation (i.e., the loss of all the phenotypes but Proliferation).

For completeness, we also compared AVATAR with MABOSS and PRISM. For this, we used GINSIM export facilities of logical models to MABOSS and PRISM formats.

MABOSS is a related command-line tool that generalizes Boolean models by defining stochastic rates associated with component updates (Stoll et al., 2012). MABOSS primary goal is to compute temporal evolutions of state probability distributions and to estimate stationary distributions. To this end, it relies on the Gillespie algorithm. MABOSS is thus well suited to get a quantitative view of temporal evolutions in the form of stochastic trajectories (see e.g., Abou-Jaoudé et al., 2016). When running MABOSS on our case studies, it appeared that the tool was able to provide the reachability probabilities of the stable states of the random models 1 to 3. However, the presence of large transient SCCs or of complex attractors hinders the evaluation of such a measure for the synthetic models and for the cell cycle model. **Table 3** includes the results obtained with MABOSS for the analysis of the **bladder tumourigenesis model**. Reachability probabilities obtained for the stable states are close to those provided by AVATAR. While MABOSS is clearly faster than AVATAR, it is unable to assess complex attractors being thus applicable only when attractors are known to be stable states.

PRISM is a model checker that supports probabilistic reachability queries (Kwiatkowska et al., 2011, 2017). To compare AVATAR and PRISM, we repeated the analysis of the **segment polarity model** with 2 cells. Results are provided in **Table 4**. Notably, PRISM is extremely efficient to evaluate the number of reachable states, a feature not provided by AVATAR. PRISM performs an exhaustive exploration to evaluate exact reachability probabilities. However, as demonstrated with AVATAR, a restricted sample of the dynamics may provide good enough probability estimates in a much shorter time. This feature is particularly useful for larger models. Indeed, for the **sp4 model**, PRISM ran out of memory and was thus unable to evaluate the number of reachable states and conclude the analysis (even when increasing the amount of available memory to CUDD to 8Gb).

5. DISCUSSION

For models of regulatory networks controlling cell fates, it is of a real interest to identify the model attractors, as well as quantify their reachability over the whole state space or from specific initial conditions. In particular, the impact of model perturbations (e.g., corresponding to observed mutations) on attractors and their basins of attraction has been used to better understand the fates of tumor cells (Huang et al., 2009; Kim et al., 2017; Shah et al., 2018). Most studies rely on Boolean models under a synchronous updating scheme. However, while stable states are identical whatever the updating scheme, it is not the case for the complex attractors, neither for the basins of all attractors. Because the synchronous scheme stems from the assumption that delays associated with component updates are equal, asynchronous updates have been considered more realistic (Thomas, 1991; Abou-Jaoudé et al., 2016). In the context of non-deterministic asynchronous dynamics, it is then relevant to assess the likelihood to reach an attractor and how model perturbations modifies this reachability likelihood. For example, this approach has been used to assess patterns of genetic alterations in bladder tumourigenesis (Remy et al., 2015), or yet to highlight the synergetic roles of Notch gain-of-function and p53 loss-of-function in promoting metastasis (Cohen et al., 2015).

Attractor identification could be achieved by analysing the State Transition Graph (STG) kept in memory but, due to combinatorial explosion, this is impractical for large models. In any case, we are still left with the problem of quantifying attractor reachability in asynchronous dynamics. As an attempt

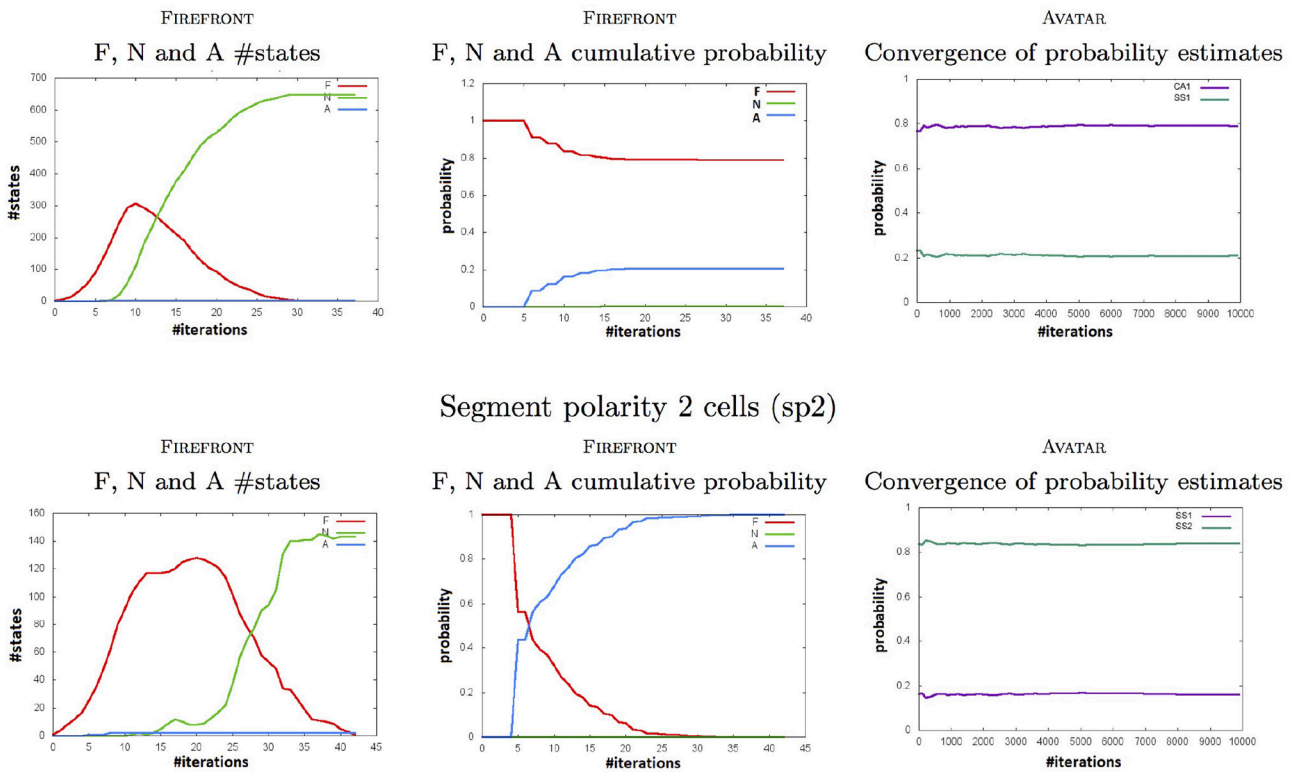


FIGURE 4 | Plots computed by FIREFRONT and AVATAR throughout simulations for the *random3* **(Top)** and the *sp1* **(Bottom)** models (see **Tables 1**, **2**). Left plots show the numbers of states to be expanded (in *F*), of neglected states (in *N*), and of attractors (in *A*). Middle plots show the cumulative probabilities of the 3 sets. Right plots show the convergence of the reachability probability of each attractor.

TABLE 3 | Attractor analysis of the bladder tumourigenesis model performed with AVATAR and MABOSS.

DNA damage	EGFR stimulus	FGFR3 stimulus	Growth inhibitor	AVATAR				MABOSS		
				Time	Attractors	Prob.	Largest SCC	Time	Attractors	Prob.
0	0	0	0	162s	GA1	1.00	163 528	15.76s	GA1	1.00
0	0	0	1	284s	GA2	0.882	239 994	15.81s	GA2	0.885
					GA3	0.118			GA3	0.115
0	0	1	0	373s	Pr1	1.00	253 440	13s	Pr1	1.00
0	0	1	1	258s	GA4	0.770	135 483	14.95s	GA4	0.722
					GA5	0.095			GA5	0.121
					Pr2	0.135			Pr2	0.157
0	1	0	0	382s	Un1 (#184 320)	1.00	184 320	699s	—	—
0	1	0	1	421s	GA6 (#512)	1.00	242 486	457.57s	—	—
0	1	1	0	212s	Pr1	1.00	151 435	11.14s	Pr1	1.00
0	1	1	1	176s	GA4	0.775	289 593	11.2s	GA4	0.737
					GA5	0.070			GA5	0.1
					Pr2	0.155			Pr2	0.162

The eight input configurations with DNA damage at 0 *are considered. Attractors are named depending on the corresponding phenotype: Pr for Proliferation, GA for Growth Arrest, Ap for Apoptosis, Un for Undecided (output components are oscillating). Attractor sizes are indicated for complex attractors. Attractor probabilities are estimated over the whole state space.* Avatar *parameters:* 10^3 *runs, up to* 10^6 *states for expansion, up to* 10^3 *states for rewiring, and* 10^8 *maximum depth.* MaBoSS *parameters: time tick*=1.0*; max time*=10^4*; sample count*=10^4*; discrete time=1.*

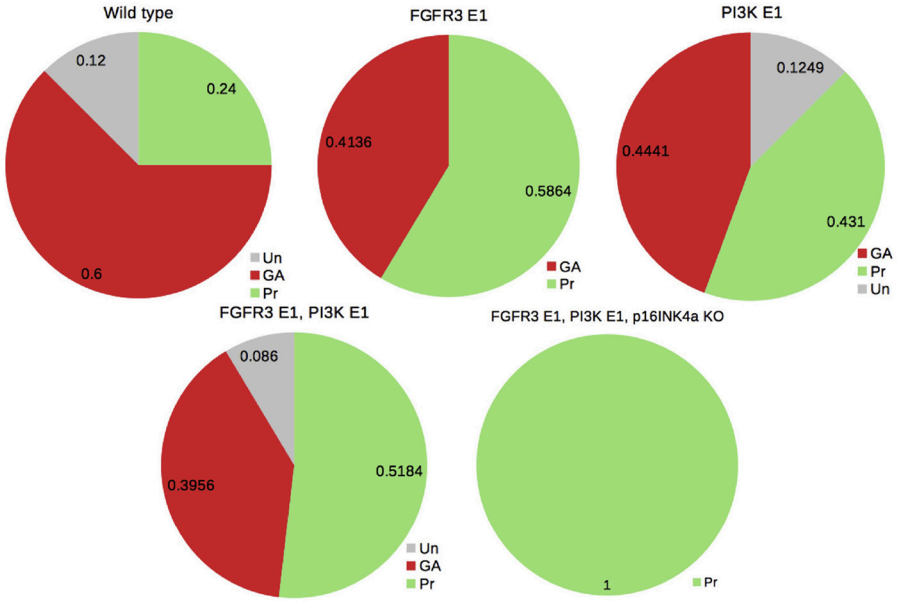


FIGURE 5 | Probabilities of the phenotypes for the bladder tumourigenesis model in the wild type and mutant contexts: probabilities for the double mutant FGRF3 overexpression (FGFR3 E1) and PI3K overexpression (PI3K E1) suggest a slight advantage in mutating FGFR3 in a PI3K-mutated context (by increasing the probability of Proliferation); a third mutation of the tumor suppressor CDKN2A (coding for p16INKa) leads to the sole phenotype Proliferation (see Remy et al., 2015).

TABLE 4 | Assessing attractor probabilities for the sp2 model with AVATAR and PRISM.

	AVATAR				PRISM	
	1E6 runs		1E4 runs			
Stable states	**Time**	**Prob.**	**Time**	**Prob.**	**Time**	**Prob.**
SS1		0.8915		0.8921		0.8909
SS2	4h59	0.1084	153s	0.1078	4h25	0.1088
SS3		1.2E-4		1E-4		1.04E-4

Probabilities returned by Avatar *are quite similar when considering a lower number of runs, indicating that it is possible to quickly obtain good estimates of reachability probabilities in a much shorter time.*

to surpass efficiency bottlenecks and quantification biases of existing methods, we have delineated two novel strategies. FIREFRONT performs a memoryless breath-first exploration of the STG, avoiding any further exploration of states which fall bellow a given threshold α. AVATAR performs a modified version of the Monte Carlo algorithm, avoiding the exploration of states previously visited by rewiring and appropriately associating new probabilities with state transitions. To adequately choose the algorithm and optimal values of associated parameters, information about the structure of the dynamics would be needed, which is generally unachievable. Broadly, the breadth of the explored STG and the structure of transient Strongly Connected Components (SCCs) clearly impact FIREFRONT's performances. AVATAR's performances are influenced by the degree of connectivity of the SCCs. Ideally, AVATAR should avoid to rewire SCCs from which it can easily exit (low connectivity or high exit ratio). On the other hand, it should rewire SCCs from which it is hard to escape. It is also much more efficient to rewire a whole SCC than to iteratively rewire portions of it. While sizes and structures of SCCs are not known *a priori*, AVATAR incorporates heuristics that evolve running parameters to the information collected in the course of the simulation.

Results from synthetic and real biological models reveal the ability of FIREFRONT and AVATAR to efficiently assess attractor reachability. This type of analysis will permit further biological insights into the dynamics of regulatory and signalling networks. For example, as mentioned above, how model perturbations modify the probability to reach an attractor can reveal the role of single or combined mutations in disease progression. Usage of both algorithms is facilitated through their implementation in GINsim, which provides a convenient graphical user interface.

As future work, the consideration of non-uniform transition probabilities could be easily handled. In particular, when priority classes can be defined by classifying component updates into e.g., slow and fast processes (Fauré et al., 2006), some trajectories are discarded thus modifying the structure of the STG, and therefore the reachability properties. Furthermore, confronting asymptotic model dynamics against experimental time series could provide the ground for model validation.

AUTHOR CONTRIBUTIONS

CC, PM, JC, and ER designed the research. CC supervised the work, PM supervised the computational implementations, and

ER focused on the theoretical foundations. NM specified the algorithms, and implemented the first prototypes. RH revised the algorithms to improve performances, and worked on the GINsim implementation. NM, RH, ER, PM, and CC wrote the paper. All authors reviewed the content of the paper and agreed to endorse it.

FUNDING

This work was supported by Fundação para a Ciência e a Tecnologia (FCT, Portugal), grants PTDC/EIA-CCO/099229/2008, UID/CEC/50021/2013, IF/01333/2013, and PTDC/EEI-CTP/2914/2014.

ACKNOWLEDGMENTS

CC would like to thank the Fundação Calouste Gulbenkian for its continuous support.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphys.2018.01161/full#supplementary-material

GINsim 3.0 is freely available at http://ginsim.org. The accompanying user documentation, algorithmic details and supplementary results will be made publicly available upon publication.

REFERENCES

Abou-Jaoudé, W., Traynard, P., Monteiro, P. T., Saez-Rodriguez, J., Helikar, T., Thieffry, D., et al. (2016). Logical modeling and dynamical analysis of cellular networks. *Front. Genet.* 7:94. doi: 10.3389/fgene.2016.00094

Calzone, L., Tournier, L., Fourquet, S., Thieffry, D., Zhivotovsky, B., Barillot, E., et al. (2010). Mathematical modelling of cell-fate decision in response to death receptor engagement. *PLoS Comput. Biol.* 6:e1000702. doi: 10.1371/journal.pcbi.1000702

Chaouiya, C., Naldi, A., and Thieffry, D. (2012). Logical modelling of gene regulatory networks with GINsim. *Methods Mol. Biol.* 804, 463–479. doi: 10.1007/978-1-61779-361-5

Chaves, M., and Preto, M. (2013). Hierarchy of models: from qualitative to quantitative analysis of circadian rhythms in cyanobacteria. *Chaos* 23:025113. doi: 10.1063/1.4810922

Cho, S.-H., Park, S.-M., Lee, H.-S., Lee, H.-Y., and Cho, K.-H. (2016). Attractor landscape analysis of colorectal tumorigenesis and its reversion. *BMC Syst. Biol.* 10:96. doi: 10.1186/s12918-016-0341-9

Cohen, D. P. A., Martignetti, L., Robine, S., Barillot, E., Zinovyev, A., and Calzone, L. (2015). Mathematical modelling of molecular pathways enabling tumour cell invasion and migration. *PLoS Comput. Biol.* 11:e1004571. doi: 10.1371/journal.pcbi.1004571

Collombet, S., van Oevelen, C., Sardina Ortega, J. L., Abou-Jaoudé, W., Di Stefano, B., Thomas-Chollier, M., et al. (2017). Logical modeling of lymphoid and myeloid cell specification and transdifferentiation. *Proc. Natl. Acad. Sci. U.S.A.* 114, 5792–5799. doi: 10.1073/pnas.1610622114

Fauré, A., Naldi, A., Chaouiya,C., and Thieffry, D. (2006). Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics* 22, 124–131. doi: 10.1093/bioinformatics/btl210

Fauré, A., and Thieffry, D. (2009). Logical modelling of cell cycle control in eukaryotes: a comparative study. *Mol. Biosyst.* 5, 1569–1581. doi: 10.1039/B907562n

Flobak, A., Baudot, A., Remy, E., Thommesen, L., Thieffry, D., Kuiper, M., et al. (2015). Discovery of drug synergies in gastric cancer cells predicted by logical modeling. *PLoS Comput. Biol.* 11:e1004426. doi: 10.1371/journal.pcbi.1004426

Garg, A., Di Cara, A., Xenarios, I., Mendoza, L., and De Micheli, G. (2008). Synchronous versus asynchronous modeling of gene regulatory networks. *Bioinformatics* 24, 1917–1925. doi: 10.1093/bioinformatics/btn336

Glass, L., and Siegelmann, H. (2010). Logical and symbolic analysis of robust biological dynamics. *Curr. Opin. Genet. Dev.* 20, 644–649. doi: 10.1016/j.gde.2010.09.005

Grinstead, C. M., Snell, J. L., and Snell, J. L. (1997). *Introduction to Probability, 2nd rev. Edition*. Providence, RI: American Mathematical Society.

Henzinger, T. A., Mateescu, M., and Wolf, V. (2009). "Sliding window abstraction for infinite markov chains," in *Computer Aided Verification*; *Lecture Notes in Computer Science*,eds A. Bouajjani and O. Maler (Berlin; Heidelberg: Springer), 337–352.

Huang, S., Ernberg, I., and Kauffman, S. (2009). Cancer attractors: a systems view of tumors from a gene network dynamics and developmental perspective. *Semin. Cell Dev. Biol.* 20, 869–876. doi: 10.1016/j.semcdb.2009.07.003

Kim, Y., Choi, S., Shin, D., and Cho, K.-H. (2017). Quantitative evaluation and reversion analysis of the attractor landscapes of an intracellular regulatory network for colorectal cancer. *BMC Syst. Biol.* 11:45. doi: 10.1186/s12918-017-0424-2

Klarner, H., Bockmayr, A., and Siebert, H. (2015). Computing maximal and minimal trap spaces of Boolean networks. *Nat. Comput.* 14, 535–544. doi: 10.1007/s11047-015-9520-7

Kwiatkowska, M., Norman, G., and Parker, D. (2011). "PRISM 4.0: Verification of probabilistic real-time systems," in *Proc. 23rd International Conference on Computer Aided Verification (CAV 2011)*; volume 6806 of *LNCS*, eds G. Gopalakrishnan and S. Qadeer (Berlin; Heidelberg: Springer), 585–591.

Kwiatkowska, M., Norman, G., and Parker, D. (2017). "Probabilistic model checking: advances and applications," in *Formal System Verification* (Cham: Springer), 73–121.

Munsky, B., and Khammash, M. (2006). The finite state projection algorithm for the solution of the chemical master equation. *J. Chem. Phys.* 124:044104. doi: 10.1063/1.2145882

Müssel, C., Hopfensitz, M., and Kestler, H. A. (2010). BoolNet- an R package for generation, reconstruction and analysis of boolean networks. *Bioinformatics* 26, 1378–1380. doi: 10.1093/bioinformatics/btq124

Naldi, A., Carneiro, J., Chaouiya, C., and Thieffry, D. (2010). Diversity and plasticity of Th cell types predicted from regulatory network modelling. *PLoS Comput. Biol.* 6:e1000912. doi: 10.1371/journal.pcbi.1000912

Naldi, A., Hernandez, C., Abou-Jaoudé, W., Monteiro, P. T., Chaouiya, C., and Thieffry, D. (2018). Logical modeling and analysis of cellular regulatory networks with ginsim 3.0. *Front. Physiol.* 9:646. doi: 10.3389/fphys.2018.00646

Naldi, A., Thieffry, D., and C. Chaouiya (2007). "Decision diagrams for the representation of logical models of regulatory networks," in *CMSB'07*, volume 4695 of *Lecture Notes in Bioinformatics (LNBI)* (Edinburgh), 233–247.

Prum, B. (2012). Chaînes de Markov et absorption. application à l'algorithme de Fu en génomique. *J. Soc. Française de Stat.* 153, 37–51. Available online at: http://journal-sfds.fr/article/view/120/110

Remy, E., Rebouissou, S., Chaouiya, C., Zinovyev, A., Radvanyi, F., and Calzone, L. (2015). A modeling approach to explain mutually exclusive and co-occurring genetic alterations in bladder tumorigenesis. *Cancer Res.* 75, 4042–4052. doi: 10.1158/0008-5472.CAN-15-0602

Saadatpour, A., and Albert, R. (2012). Discrete dynamic modeling of signal transduction networks. *Methods Mol. Biol.* 880, 255–272. doi: 10.1007/978-1-61779-833-7

Sánchez, L., Chaouiya, C., and Thieffry, D. (2008). Segmenting the fly embryo: logical analysis of the role of the Segment Polarity cross-regulatory module. *Int. J. Dev. Biol.* 52, 1059–1075. doi: 10.1387/ijdb.072439ls

Shah, O. S., Chaudhary, M. F. A., Awan, H. A., Fatima, F., Arshad, Z., Amina, B., et al. (2018). ATLANTIS - Attractor landscape analysis toolbox for cell fate discovery and reprogramming. *Sci. Rep.* 8:3554. doi: 10.1038/s41598-018-22031-3

Stoll, G., Viara, E., Barillot, E., and Calzone, L. (2012). Continuous time Boolean modeling for biological signaling: application of gillespie algorithm. *BMC Syst. Biol.* 6:116. doi: 10.1186/1752-0509-6-116

Tarjan, R. (1972). Depth-first-search and linear graph algorithms. *SIAM J. Comput.* 1, 146–160.

Thomas, R. (1991). Regulatory networks seen as asynchronous automata: a logical description. *J. Theor. Biol.* 153, 1–23.

Thomas, R., and d'Ari, R. (1990). *Biological Feedback*. Boca Raton, FL: CRC Press.

Zañudo, J. G. T., and Albert, R. (2013). An effective network reduction approach to find the dynamical repertoire of discrete dynamic networks. *Chaos* 23:025111. doi: 10.1063/1.4809777

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

ORIGINAL RESEARCH
published: 28 September 2018
doi: 10.3389/fphys.2018.01364

Dynamics of the Gene Regulatory Network of HIV-1 and the Role of Viral Non-coding RNAs on Latency Reversion

Antonio Bensussen[1†], Christian Torres-Sosa[1,2,3†], Ramón A. Gonzalez[4] and José Díaz[1*]*

[1] Laboratory of Gene Networks Dynamics, Centro de Investigación en Dinámica Celular, Universidad Autónoma del Estado de Morelos, Cuernavaca, Mexico, [2] Centro de Ciencias de la Complejidad, Universidad Nacional Autónoma de México, Ciudad de México, Mexico, [3] Instituto de Ciencias Físicas, Universidad Nacional Autónoma de México, Cuernavaca, Mexico, [4] Laboratory of Molecular Virology, Centro de Investigación en Dinámica Celular, Universidad Autónoma del Estado de Morelos, Cuernavaca, Mexico

The use of latency reversing agents (LRAs) is currently a promising approach to eliminate latent reservoirs of HIV-1. However, this strategy has not been successful *in vivo*. It has been proposed that cellular post-transcriptional mechanisms are implicated in the underperformance of LRAs, but it is not clear whether proviral regulatory elements like viral non-coding RNAs (vncRNAs) are also implicated. In order to visualize the complexity of the HIV-1 gene expression, we used experimental data to construct a gene regulatory network (GRN) of latent proviruses in resting CD4+ T cells. We then analyzed the dynamics of this GRN using Boolean and continuous mathematical models. Our simulations predict that vncRNAs are able to counteract the activity of LRAs, which may explain the failure of these compounds to reactivate latent reservoirs of HIV-1. Moreover, our results also predict that using inhibitors of histone methyltransferases, such as chaetocin, together with releasers of the positive transcription elongation factor (P-TEFb), like JQ1, may increase proviral reactivation despite self-repressive effects of vncRNAs.

Keywords: HIV-1, viral non-coding RNAs, reservoirs, antiretroviral therapy, LRAs, dynamics, Boolean networks

OPEN ACCESS

Edited by:
Matteo Barberis, University of Amsterdam, Netherlands

Reviewed by:
Pengyue Zhang, Indiana University Bloomington, United States
Dimiter Prodanov, Interuniversity Microelectronics Centre (IMEC), Belgium

***Correspondence:**
Antonio Bensussen
bensussenantonio@gmail.com
José Díaz
biofisca@yahoo.com

†These authors have contributed equally to this work

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: *29 March 2018*
Accepted: *07 September 2018*
Published: *28 September 2018*

Citation:
Bensussen A, Torres-Sosa C, Gonzalez RA and Díaz J (2018) Dynamics of the Gene Regulatory Network of HIV-1 and the Role of Viral Non-coding RNAs on Latency Reversion. Front. Physiol. 9:1364. doi: 10.3389/fphys.2018.01364

INTRODUCTION

Combined antiretroviral therapy (cART) is currently the most effective approach to control the chronic infection of HIV-1. However, cART does not eliminate the virus even with treatment intensification (Dinoso et al., 2009). This occurs because HIV-1 is able to form long-lived reservoirs by remaining latent within resting memory CD4+ T-cells (Siliciano et al., 2003; Siliciano and Greene, 2011; Cohn et al., 2015). Recently it has been proposed the use of LRAs in combination with cART to eliminate latently infected cells. Ideally this "*shock-and-kill*" strategy could purge viral reservoirs because when LRAs reactivate latently infected cells, those cells may be eliminated by self HIV-1 replication or by action of the immune system while cART prevents the formation of new viral reservoirs (Deeks, 2012). Despite many *in vitro* observations suggest that this strategy can be a promising approach (Deeks, 2012), clinical trials with LRAs have shown that it is ineffective *in vivo* (Bullen et al., 2014). Stochastic modeling of latently infected cells indicated that the clinical underperformance of LRAs is due to their inability to minimize the size of the viral reservoirs (Hill et al., 2014). Furthermore, this study suggested that LRAs must reduce the size of viral reservoirs

10,000-fold to prevent HIV rebounds after cART (Hill et al., 2014), an objective that cannot be reached with current treatments (Cillo et al., 2014).

The "*shock-and-kill*" strategy is based on the assumption that proviral reactivation depends only on the immunological activation of the infected cells. However, recent findings suggest that this assumption is not entirely true, since it has been observed that the provirus is able to autonomously regulate its latency using the positive feedback loop of trans-activator of transcription (Tat) independently of cell activation (Razooky et al., 2015). During early stages of infection, Tat is synthesized at low levels that fluctuate because of cell's downregulation of the provirus (Weinberger and Shenk, 2007). When these transcriptional fluctuations are sustained, the activity of Tat initiates a positive feedback loop which boosts proviral transcription by recruiting P-TEFb in order to increase the synthesis of full-length viral RNAs (Weinberger and Shenk, 2007; Romani et al., 2010). In a biological context the two classical functions of positive feedback loops are to amplify and to sustain gene expression (Zhang Q. et al., 2014), however the architecture of the Tat circuit only amplifies transcriptional fluctuations making the gene expression of provirus transitory (Weinberger et al., 2005; Weinberger and Shenk, 2007). This architecture constitutes a mechanism of negative self-regulation of HIV-1, which may hinder viral reactivation (Razooky et al., 2015), and therefore may obstruct the activity of LRAs. Nevertheless, Tat is not the only structural component of HIV-1 that has a regulatory circuit. It has been observed that several vncRNAs have their own positive and negative feedback loops that may increase or suppress gene expression of the provirus (Groen and Morris, 2013; Saayman et al., 2014; Zhang Y. et al., 2014; Suzuki et al., 2015). It has been suggested that those vncRNAs have a secondary role on latency maintaining (Suzuki et al., 2015) and it is not clear whether such viral components participate in the low efficiency of the LRAs.

Current mathematical models of HIV-1 biology have been focused on transmission dynamics, posttreatment control, Vorinostat, and Romidepsin treatments, as well as the relation between reservoir size and reactivation (Hernandez-Vargas, 2017). However, none of these models addressed whether exist other paths to manipulate molecular components of the HIV to enhance latency reversion. Here we used Boolean and ordinary differential equations (ODEs) models to analyze the dynamics of the GRN of provirus to investigate how to reactivate more efficiently viral reservoirs with LRAs. In this network we included the interactions mediated by early viral proteins, vncRNAs, and epigenetic factors that regulate latency in resting CD4+ T-cells (**Figure 1**). It is important to remark that we used two different mathematical models in order to obtain results that represent the real dynamics of the GRN, independently of the model type chosen. The discrete model was used to calculate global properties of the network (attractors and its basins). The continuous model was used to measure changes in RNAs and protein expression levels of the GRN components. Both models consistently showed that the architecture of the GRN of wild type proviruses favors latency over activation state because of redundant interactions of vncRNAs. Furthermore, the models showed that reactivating effects of LRAs also stimulate the increase of vncRNAs, which reduces proviral protein expression. Finally, the models showed that the use of inhibitors of histone methyltransferases (HMTs) with releasers of P-TEFb, like chaetocin and JQ1 respectively, may increase proviral reactivation even in presence of vncRNAs.

Abbreviations: Antagomirs, Antagonic Micro-RNAs; ASK1, Apoptosis signal-regulating kinase 1; asRNA, Antisense RNA; cART, Combined Antiretroviral Therapy; GRN, Gene Regulatory Network; HATs, Histone Acetyltransferases; HDACis, Histone Deacetylases Inhibitors; HDACs, Histone Deacetylases; HMTis, Histone Methyltransferases Inhibitors; HMTs, Histone Methyltransferases; LRAs, Latency Reversing Agents; Nef, Negative Effector; NF-κB, Nuclear Factor κB; P-TEFb, Positive Transcription Elongation Factor; Tat, Trans-activator of Transcription; TNF, Tumor Necrosis Factor; Vif, Viral Infectivity Factor; vncRNAs, Viral Non-coding RNAs; Vpr, Viral Protein R; vsaRNA, Viral Small Activator RNA; vsiRNA, Viral Small Interfering RNA.

MATERIALS AND METHODS

This work was performed in four stages: (1) Defining the GRN and its models, (2) Mathematical analysis of the GRN models, (3) Perturbation analysis of the models, and (4) Validation. The complete flux diagram of the methodology of this work is shown in **Figure 2**. During the first stage we constructed the GRN as well as the Boolean and the continuous models, then both models were analyzed separately. For the Boolean model, it was calculated its attractors with their respective attraction basins, then it was calculated the activation trajectory of the GRN and finally, it was evaluated the sensitivity of the model with the Derrida Test. On the other hand, it was calculated the equilibria of the ODEs model and it was evaluated the behavior of trajectories around such points with the analysis of stability, it was then evaluated the effect of particular changes in parameters values with the bifurcation analysis and finally it was evaluated the sensitivity of the model with a global sensitivity analysis. In the third stage it was performed a screening assay to find perturbations that reactivate latent proviruses and it was analyzed the dynamical features of such perturbations with discrete and continuous models. Finally, we validated both models with experimental data available from literature. In the following paragraphs of this section we present details of the protocols used in this work.

Construction of the Network

The GRN was built by compiling information from the literature on the molecular mechanisms that regulate HIV-1 latency inside resting CD4+ T-cells (**Figure 1**). This GRN included the main interactions of antisense long-non coding RNAs (asRNA), viral small interfering RNAs (vsiRNA), viral small activator RNA (vsaRNA), Tat, Rev, Nef, Vpr, and cellular factors that control gene expression of latent proviruses such as histone deacetylases (HDACs), histone acetyltransferases (HATs), and HMTs.

We incorporated to the GRN the most important molecules and viral components involved in the regulation of provirus gene expression, namely: the concentration of NF-κB, HATs, and HMTs; the activity of viral promoters 5′LTR and

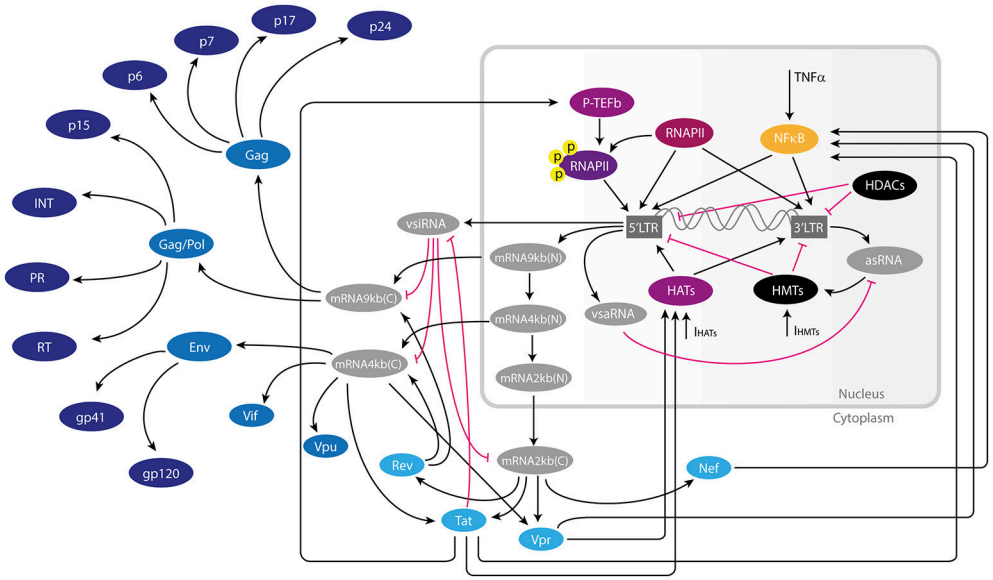


FIGURE 1 | Gene regulatory network of HIV-1 provirus. Inside latently infected resting CD4+-T cells, the provirus can be activated by transcription factors and chromatin remodeling machinery of the host during immunological stimulation. When this occurs, the provirus expresses a wide variety of RNAs such as spliced viral mRNAs, as well as non-coding viral RNAs such as vsiRNAs (viral small interfering RNAs) and asRNA (long anti-sense RNA). The non-coding RNAs together with early proteins Tat, Rev, Nef, and Vpr have regulatory functions of the provirus either inducing or repressing viral transcription. Once the intracellular conditions are favorable to viral proliferation, late proteins like gp41, p24Gag, and other structural viral proteins are produced. In this figure the repressive interaction of this network are represented by pink T-bars, and activating interactions are represented by black arrows. Early and late proteins are shown in light and dark blue, respectively. Components of the host's transcription machinery are shown in purple, yellow, and black.

3′LTR; nuclear genomic mRNA of 9 kb, [mRNA9kb(N)]; vsiRNA; vsaRNA; nuclear mRNAs of 4 kb [mRNA4kb(N)] and 2 kb [mRNA2kb(N)]; cytoplasmic genomic mRNA of 9 kb [mRNA9kb(C)], and cytoplasmic mRNAs of 4 kb [mRNA4kb(C)]; and 2 kb [mRNA2kb(C)]; as well as Tat, Rev, Nef, Vpr, asRNA, and the p24 gag protein (p24Gag). Based on the above, we proposed discrete and ODE-based mathematical models to understand the dynamical properties of the GRN. In what follows we present first the discrete model and then the continuous model.

DISCRETE MODEL

For the discrete dynamics, the state of the nodes of the network in **Figure 1** are represented by a set of binary variables, $\Sigma = \{\sigma_1, \cdots, \sigma_N\}$, each one taking the value 1 for activation and 0 for inactivation. The value of each variable σ_n is determined by its k_n regulators, denoted by $\{\sigma_{n_1}, \cdots, \sigma_{n_{k_n}}\}$, through the equation

$$\sigma_n(t + \Delta t) = f_n\left(\sigma_{n_1}(t'), \sigma_{n_2}(t'), \ldots, \sigma_{n_k}(t')\right), \qquad (1)$$

where f_n is a Boolean function that depends on k_n arguments (**Table 1**). This function is constructed according to the inhibitory or activating nature of the interactions between σ_n and its regulators (Kauffman, 1969). The discrete time t advances in integer steps; the time t' at which the state of the regulators is evaluated is such that $t \leq t' < t + \Delta t$, where Δt is the time it takes to σ_n to respond to a change in its regulators. Traditionally, Equation (1) is implemented simultaneously (synchronously) on all the nodes of the network. In this synchronous case $t' = t$ and $\Delta t = 1$. In addition to the synchronous update, we also implemented two other updating schemes: asynchronous and semi-synchronous.

In the asynchronous scheme a permutation with repetition of the network nodes $\{\sigma_1, \cdots, \sigma_N\}$ is chosen. Let us denote as $P = \{\sigma_{p_1}, \sigma_{p_2} \cdots \sigma_{p_L}\}$ this permutation, where $L \geq N$. Then at each time step t the nodes of the network are updated one by one following the order of this permutation: first σ_{p_1} at time $t' = t + \frac{1}{L}$, then σ_{p_2} at time $t' = t + \frac{2}{L}$, and so on until σ_{p_L} is updated at time $t' = t+1$. When σ_{p_i} is being updated, Equation (1) is applied with $\Delta t = \frac{i}{L}$ and $t' = t + \frac{i-1}{L}$. After all the nodes in the permutation have been updated, the time t advances one unit and the process is repeated until an attractor is reached.

For the semi-synchronous scheme the set of all network nodes $\Sigma = \{\sigma_1, \cdots, \sigma_N\}$ is partitioned into S subsets $\{M_1, \cdots, M_s\}$ such that $\bigcup_{j=1}^{S} M_j = \Sigma$. All the nodes contained in M_j are updated synchronously, but the subsets $\{M_1, \cdots, M_s\}$ are updated asynchronously: the nodes in M_1 are updated at time $t' = t + \frac{1}{S}$, the nodes in M_2 are updated at time $t' = t + \frac{2}{S}$, and son on until the nodes in M_S are updated at time $t' = t + 1$. When the nodes in M_i are being updated, Equation (1) is applied with $\Delta t = \frac{i}{S}$ and $t' = t + \frac{i-1}{S}$. A full time step to go from Δt to $t + 1$ consists in the updating of all the subsets $\{M_1, \cdots, M_s\}$, one by one in successive order. The construction of the permutation P for the asynchronous scheme and the subsets $\{M_1, \cdots, M_s\}$ for the semi-synchronous one was based

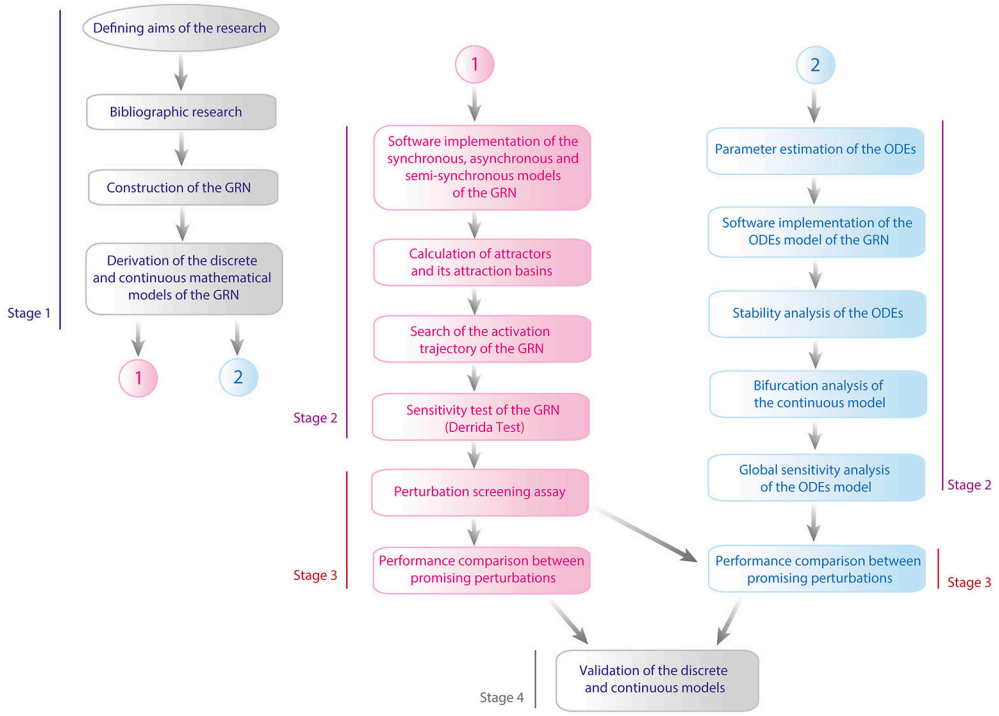


FIGURE 2 | Flux diagram of methodology. This work was executed in four stages which are: (1) Defining the GRN and its models, (2) Mathematical analysis of the models, (3) Perturbation analysis of the models, and (4) Validation. In the figure line blue denoted the first stage, line purple corresponds to stage 2, line red is for stage 3, and finally gray line is assigned to the fourth stage. Pink color is used to represent the methodological steps made with the discrete model and blue color for continuous model.

on biological phenomenology that reflects the way in which the activation cascade across the network may occur, and it is presented in the **Supplementary Material**.

It is well-known that the size of the basin of attraction is modified by updating scheme (Gershenson, 2002). The belonging of a network state to a particular basin of attraction strongly depends to updating scheme chosen. This has a biological equivalence, because the cellular environment is noisy and the order of gene expression may occur in different ways. However, there are some network states that always belong to same basin of attraction independently of updating scheme used. We call to this property as *robustness under updating scheme*. We hypothesize that the set of network states with this property are relevant for the biological behavior of the provirus. We call this set of states as *intersection of the network states*. We calculated the intersection of the synchronous, semi-synchronous, and asynchronous to determine the trajectory of activation of the provirus.

Stability of the Boolean Model: Derrida Map Test

The discrete model can exhibit two dynamical regimes, ordered and chaotic, and a phase transition between them, the so-called critical point (Aldana, 2003). The characterization of these regimes is given by the behavior of the avalanche of perturbations (produced by stochastic fluctuations, gene knockout, or gene over expression). In the chaotic regime, small perturbations spread throughout the network over time, producing big changes in the network state. Therefore, a network operating in a chaotic regime and submerged in a noisy cellular environment would have very unstable phenotypes. In the order regime, the perturbations die out over time, preventing the network to respond to new changing environmental conditions. In the critical point, the perturbations neither spread to the entire network nor disappear. They typically remain confined within a small fraction of genes. In order to characterize the dynamical regime, we define the normalized Hamming distance $h(t)$ at time t between two network states as:

$$h\left(t\right)=\frac{1}{N}\sum_{n=1}^{N}\left|\sigma_n\left(t\right)-\tilde{\sigma}_n\left(t\right)\right|. \tag{2}$$

In this equation $\sigma_n(t)$ is the state of the nth gene at time t in a trajectory starting out from a given initial condition, and $\tilde{\sigma}_n(t)$ is the state of the same gene in a different trajectory generated from a different initial condition. The Hamming distance $h(t)$ can be considered as the normalized size of the avalanche of perturbations generated by differences the two initial conditions. The Derrida map $h(t+1) = M(h(t))$ (Derrida and Pomeau, 1986) relates the size of the avalanche at two consecutive time steps. It can be shown that $M(h)$ is a monotonic increasing function with the property that $M(0) = 0$ (if there is no

TABLE 1 | Logic rules that models the GRN.

Node		Logic rule
TNFα	=	*input*
I_{HMTS}	=	*input*
NFκB	=	(*TNFα*) *OR* (*Tat*) *OR* (*Vpr*) *OR* (*Nef*)
HMTs	=	(I_{HMTS}) *OR* (*asRNA*)
p′5LTR	=	(*NFkB*) *AND NOT* (*HMTs*)
p′3LTR	=	(*NFkB*) *AND NOT* (*HMTs*)
RNAs9kbN	=	*p′5LTR*
vsiRNA	=	(*p′5LTR*) *AND NOT* (*Tat*)
vsaRNA	=	(*p′5LTR*)
RNAs4kbN	=	*RNAs9kbN*
RNAs2kbN	=	*RNAs4kbN*
RNAs2kbC	=	(RNAs2kbN) AND NOT (vsiRNA)
RNAs4kbC	=	(RNAs4kbN) AND (Rev) AND NOT (vsiRNA)
RNAs9kbC	=	(RNAs9kbN) AND (Rev) AND NOT (vsiRNA)
asRNA	=	(p'3LTR) AND NOT (vsaRNA)
Tat	=	(RNAs2kbC) OR (RNAs4kbC)
Rev	=	RNAs2kbC
Nef	=	RNAs2kbC
Vpr	=	(RNAs2kbC) OR (RNAs4kbC)
p24Gag	=	RNAs9kbC

perturbation at time t, there is no perturbation either at time $(t+1)$. The slope S at the origin of $M(h)$ is the parameter that characterizes the asymptotic value of the Hamming distance, and hence the network dynamics. S is called the average network sensitivity. When $S < 1$ the network is operating in the ordered regime. If $S > 1$, the network exhibits chaotic behavior. If $S = 1$, the network is at the critical point. An intuitive definition (Krawitz and Shmulevich, 2007) is that S is the average fraction of genes that change their state at time $t+1$ when a single gene is perturbed at time t (**Supplementary Material**). Therefore, to determine the stability of the network dynamics under perturbations in the initial conditions, one has to compute the network sensitivity S from the Derrida map $M(h)$.

Additionally, one can compute the network stability under *permanent perturbations*. We implemented two types of permanent perturbations: inhibition and overstimulation. For this, we set the state of one node, say σ_j, equal to 0 or 1 all the time (regardless of the state of its regulators). Setting $\sigma_j = 0$ for all time is equivalent to permanently inhibit this node, while setting $\sigma_j = 1$ all the time is equivalent to having this node being constantly expressed. Let us denote as S_j the network sensitivity when σ_j is permanently perturbed (either inhibited or overstimulated), and as S_0 the sensitivity of the wildtype network. In order to compare the dynamical properties of perturbed proviruses vs. the WT provirus, we define the difference of sensitivity ΔS as:

$$\Delta S = S_j - S_0. \tag{3}$$

This quantity measures how the network dynamics changes when one of the nodes is permanently perturbed. We performed the same type of analysis for the case in which two nodes σ_i and σ_j are simultaneously perturbed in a permanent way, either inhibiting or overstimulating them. This allows us to determine whether between-node epistasis exists that can modify the dynamics of the GRN.

Probability of Viral Activation

It is important to note that in the three updating schemes presented here, i.e., synchronous, asynchronous, and semi-synchronous, the network dynamics are deterministic (both the permutation P and the subsets $\{M_1, \cdots, M_s\}$ are fixed). Therefore, in any of these updating schemes, after a transient time the network will fall into an attractor (a periodic pattern of activity). Several attractors may exist, and all the initial conditions that eventually fall into the same attractor are known as the basin of attraction of that attractor. As we show in the Results section, the HIV-1 network has several attractors. In some of them the network dynamics correspond to an active virus (the viral proteins are expressed, particularly p24Gag), whereas in the other attractors the dynamics correspond to an inactive virus (i.e., in the latency state with no expression of p24Gag). We refer to the former as the *active attractors* and to the latter as the *inactive attractors*. In order to determine the probability that a given initial condition leads to the active viral state, we compute the relative size of the activation state (W_{on}) by adding the size of the basins of attraction for all active attractors and dividing this sum by the total number of network states:

$$W_{on} = \frac{1}{\Omega} \sum_k |B(a_k)|, \tag{4}$$

where $\Omega = 2^N$ is the total amount of network states, and $|B(a_k)|$ is the size of the basin of attraction of the k–th active attractor. Similarly, the relative size of latency state (W_{off}) was calculated as follows:

$$W_{off} = 1 - W_{on}. \tag{5}$$

These metrics determine the frequency of each state of the GRN that leads to an active or inactive attractor.

CONTINUOUS MODEL

In the continuous model, we represent the state of the nodes of the network in **Figure 1** by the continuous variables $\{x_1, \cdots, x_N\}$, which satisfy the general equation of mass balance (**Table 2**)

$$\frac{dx_n}{dt} = \sum_k J^i_{n_k} - \sum_j J^o_{n_j}, \tag{6}$$

where the sums $\sum_k J^i_{n_k}$ and $\sum_j J^o_{n_j}$ represent all the fluxes that contribute to increase and decrease x_n, respectively. The fluxes are presented in detail in **Table 2**, and the kinetic parameters (which were obtained from the literature), in the **Supplementary Material**. The Runge-Kutta 4-5 method was used to solve the system of ODEs.

TABLE 2 | Ordinary Differential Equations that models the GRN.

Node	Equation	Fluxes
5′LTR	$\dot{J}_{5LTR} = J_1 - J_2$	$J_1 = k_b\left(1 + k_{ac}\left[HATs\right] + k_{tar}\left[Tat\right]\right)\left(\left[RNAP\right]_T - \left[3LTR\right] - \left[5LTR\right]\right)\left[NFkB\right]$ $J_2 = k_d\left(1 + k_{me}\left[HMTs\right]\right)\left[5LTR\right]$
3′LTR	$\dot{J}_{3LTR} = J_3 - J_4$	$J_3 = k_b\left(1 + k_{ac}\left[HATs\right]\right)\left(\left[RNAP\right]_T - \left[3LTR\right] - \left[5LTR\right]\right)\left[NFkB\right]$ $J_4 = k_d\left(1 + k_{me}\left[HMTs\right]\right)\left[3LTR\right]$
RNA9kbN	$\dot{J}_{9kbC} = J_5 - J_6 - J_7 - J_8$	$J_5 = a_1\left[5LTR\right]$ $J_6 = \left(s_1 + \tau + k_{RRE}\left[Rev\right]\right)\left[RNAs9kbN\right]$ $J_7 = \delta_1\left[RNAs9kbN\right]$ $J_8 = s_1\left[RNAs9kbN\right]$
vsiRNA	$\dot{J}_{vsiRNA} = J_9 - J_{10}$	$J_9 = a_2\left[RNA_{9kb}\right]$ $J_{10} = \left(\delta_2 + r_1\left[Tat\right]\right)\left[vsiRNA\right]$
vsaRNA	$\dot{J}_{vsaRNA} = J_{11} - J_{12}$	$J_{11} = a_3\left[RNA_{9kb}\right]$ $J_{12} = \delta_3\left[vsaRNA\right]$
asRNA	$\dot{J}_{asRNA} = J_{13} - J_{14}$	$J_{13} = a_4\left[3LTR\right]$ $J_{14} = \left(\delta_4 + r_2\left[vsaRNA\right]\right)\left[asRNA\right]$
RNA4kbN	$\dot{J}_{4kbC} = J_8 - J_{15} - J_{16} - J_{17}$	$J_{15} = \left(\tau + k_{RRE}\left[Rev\right]\right)\left[RNAs4kbN\right]$ $J_{16} = \delta_1\left[RNAs4kbN\right]$ $J_{17} = s_2\left[RNAs4kbN\right]$
RNA2kbN	$\dot{J}_{2kbN} = J_{17} - J_{18} - J_{19}$	$J_{18} = k_{\exp}\left[RNAs2kbN\right]$ $J_{19} = \delta_6\left[RNAs2kbN\right]$
RNA2kbC	$\dot{J}_{2kbC} = J_{18} - J_{20} - J_{21}$	$J_{20} = \delta_7\left[RNAs2kbC\right]$ $J_{21} = r_3\left[vsiRNA\right]\left[RNAs2kbC\right]$
RNA4kbC	$\dot{J}_{4kbC} = J_{15} - J_{22} - J_{23}$	$J_{22} = \delta_8\left[RNAs4kbC\right]$ $J_{23} = r_3\left[vsiRNA\right]\left[RNAs4kbC\right]$
RNA9kbC	$\dot{J}_{9kbC} = J_6 - J_{24} - J_{25}$	$J_{24} = \delta_9\left[RNAs9kbC\right]$ $J_{25} = r_3\left[vsiRNA\right]\left[RNAs9kbC\right]$
Tat	$\dot{J}_{Tat} = J_{26} + J_{27} - J_{28}$	$J_{26} = a_5\left[RNAs2kbC\right]$ $J_{27} = a_6\left[RNAs4kbC\right]$ $J_{28} = \delta_{10}\left[Tat\right]$
Rev	$\dot{J}_{Rev} = J_{29} - J_{30}$	$J_{29} = a_7\left[RNAs2kbC\right]$ $J_{30} = \delta_{11}\left[Rev\right]$
Nef	$\dot{J}_{Nef} = J_{31} - J_{32}$	$J_{31} = a_8\left[RNAs2kbC\right]$ $J_{32} = \delta_{12}\left[Nef\right]$
Vpr	$\dot{J}_{Vpr} = J_{33} + J_{34} - J_{35}$	$J_{33} = a_9\left[RNAs2kbC\right]$ $J_{34} = a_{10}\left[RNAs4kbC\right]$ $J_{35} = \delta_{14}\left[Vpr\right]$
p24Gag	$\dot{J}_{p24Gag} = J_{36} - J_{37}$	$J_{36} = a_{11}\left[RNAs9kbC\right]$ $J_{37} = \delta_{15}\left[p24Gag\right]$
NF-kB	$\dot{J}_{NFkB} = J_{38} - J_{39}$	$J_{38} = k_1\left(\left[NFkB\right]_T - \left[NFkB\right]\right)\left(k_0\left[TNF\right] + k_2\left[Tat\right] + k_3\left[Nef\right] + k_4\left[Vpr\right]\right)$ $J_{39} = k_{-1}\left[NFkB\right]$
HATs	$\dot{J}_{HATs} = J_{40} - J_{41}$	$J_{40} = k_5\left(\left[HATs\right]_T - \left[HATs\right]\right)\left(l_{HATs} + k_6\left[Tat\right] + k_7\left[Vpr\right]\right)$ $J_{41} = k_{-5}\left[HATs\right]$
HMTs	$\dot{J}_{HMTs} = J_{42} - J_{43}$	$J_{42} = k_8\left(\left[HMTs\right]_T - \left[HNTs\right]\right)\left(l_{HMTs} + k_9\left[asRNA\right]\right)$ $J_{43} = k_{-8}\left[HMTs\right]$

Input Signals for the GRN

The transcriptional state of provirus can be modified by the NF-κB pathway activated by the *Tumor Necrosis Factor* (TNF) and by chromatin modifications such as acetylation and methylation (**Supplementary Material**). Those modifications are produced by HMTs and HATs in response to intracellular stimulator signals, represented by I_{HMTs} and I_{HATs}, respectively. We take TNF, I_{HMTs}, and I_{HATs} as the inputs of the GRN. In the Boolean model these inputs have only two states $\{0, 1\}$, which are inactivation and activation respectively. In the ODEs model we use square pulse functions to model the inputs as follows:

For extracellular pulses of TNF:

$$TNF\left(t\right) = \begin{cases} 1, & t \in T_1 \\ 0, & t \notin T_1 \end{cases} \tag{7}$$

For signals that stimulate HATs activity:

$$I_{HATs}\left(t\right) = \begin{cases} 1, & t \in T_2 \\ 0, & t \notin T_2 \end{cases} \tag{8}$$

For signals that stimulate HMTs activity:

$$I_{HMTs}\left(t\right) = \begin{cases} 1, & t \in T_3 \\ 0, & t \notin T_3 \end{cases} \tag{9}$$

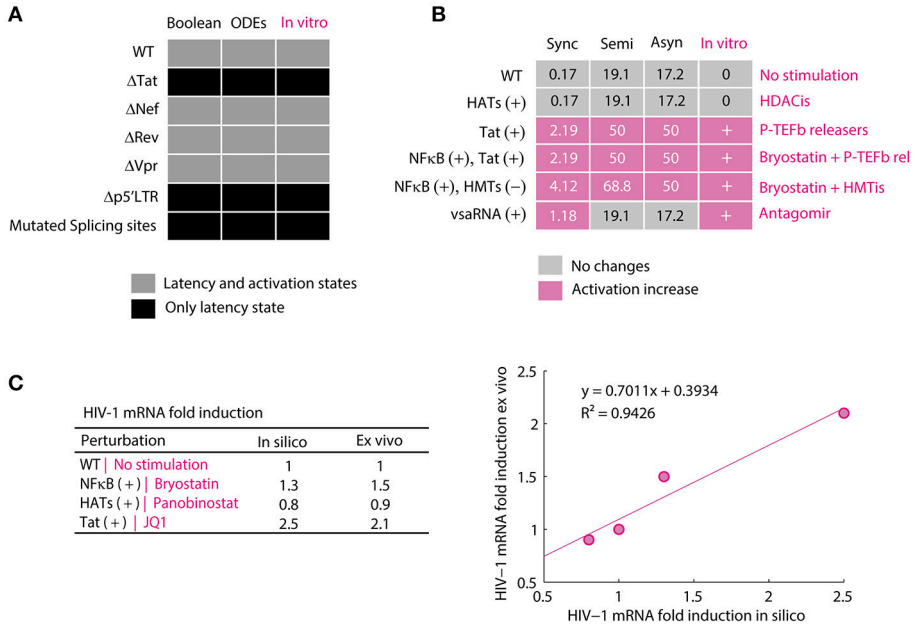


FIGURE 3 | Validation of mathematical models of the HIV-1 GRN. **(A)** Compatibility of the models. In this panel were qualitatively compared the attractors of the Boolean model and the equilibrium points of the ODEs model to provirus behavior observed *in vitro*. The discrete and continuous models present activation and latency states for WT proviruses and deletions of *nef* and *vpr* but only present attractors and equilibrium points for latency state when tat, p5′LTR and splicing sites. This behavior is the same as reported for defective p′5LTR mutants (Ho et al., 2013) and the splicing sites (Purcell and Martin, 1993), deleted *tat* (Verhoef and Berkhout, 1999), *vpr* (Rücker et al., 2004), *rev* and *nef* (Churchill et al., 2007). **(B)** Validation of the Boolean model. In this panel is shown the size of activation state (W_{on}) calculated with the synchronous, semi-synchronous, and asynchronous update scheme. In pink is shown increases of W_{on} with respect to WT provirus. In the column of *in vitro* observations, "+" represents that there was an increase of viral reactivation because of the treatment and "0" indicates that there were no changes. The data for HDACis was obtained from (Cillo et al., 2014), for P-TEFb releasers from (Li et al., 2013), the use of Antagomirs from (Zhang Y. et al., 2014), combinations of Bryostatin with P-TEFb releasers from (Laird et al., 2015) and combinations of Bryostatin with HMTis from (Bouchat et al., 2012). **(C)** Validation of the ODEs model. In this panel is presented the normalized data of unspliced viral mRNAs levels obtained with the ODEs model and the corresponding values obtained from patients treated with bryostatin (Bullen et al., 2014), panobinostat (Laird et al., 2015), and JQ1 (Laird et al., 2015) vs. their corresponding simulation. Pearson correlation between both data sets showed a positive linear relationship, $p = 0.0291$, $r_{(3)} = 0.9708$, which supports the validity of the model. The standard error of linear regression was 0.1613.

In these equations T_1, T_2, and T_3 are the activation intervals of the input signals (**Supplementary Material**).

Stability Analysis

The stability analysis of the continuous system was performed using the indirect method of Lyapunov (**Supplementary Material**). This method starts solving the ODEs in order to find the equilibrium points of the system. Then the ODEs are linearized using the Jacobian matrix to calculate the eigenvalues for all equilibrium points (**Supplementary Material**). Positive eigenvalues correspond to unstable directions in the phase space, whereas negative eigenvalues correspond to stable directions. If all the eigenvalues corresponding to one equilibrium point are negative, then that point is stable.

Bifurcation Analysis

The bifurcation analysis of the ODEs model was performed by changing one by one the parameters of the model. We focused our attention on the dissociation constants of NF-κB, association and dissociation constants of viral proteins, and degradation constants of RNA's and viral proteins (**Supplementary Material**). Then, each parameter was varied three orders of magnitude, up and down of their reference value and after that; MATLAB was used to calculate the equilibrium points of the system with their corresponding stability.

Global Sensitivity Analysis

The sensitivity of the model against random perturbations was evaluated by assigning a uniform distribution to each parameter in which their reference value was taken as the mean and the standard deviation was assumed to be 10% (**Supplementary Material**). Then, each distribution was randomly sampled to obtain a set of parameters that were used as the inputs to solve the equations of the model during 1,500 units of time. After 10,000 iterations of this process, the concentration of p24Gag was used as the system's output to analyze the behavior of the model in response to random parameter variation. In all the simulations we set $TNF(t) = 0$, $I_{HATs}(t) = 0$ and $I_{HMTs}(t) = 0$.

Simulating Mutants and Treatments

The behavior of mutant proviruses during the condensation of viral nucleosomes and T-cells activation was modeled by

TABLE 3 | Attractors of the HIV Boolean model.

Nodes	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}
TNF	0	1	0	1	0	1	0	1	0	1	0	1
IHATs	0	0	1	1	0	0	1	1	0	0	1	1
IHMTs	0	0	0	0	1	1	1	1	0	0	0	0
NF-κB	0	1	0	1	0	1	0	1	1	1	1	1
HATs	0	0	1	1	0	0	1	1	1	1	1	1
HMTs	0	0	0	0	1	1	1	1	0	0	0	0
p5'LTR	0	1	0	1	0	0	0	0	1	1	1	1
p3'LTR	0	1	0	1	0	0	0	0	1	1	1	1
mRNA9kb(N)	0	1	0	1	0	0	0	0	1	1	1	1
vsiRNA	0	1	0	1	0	0	0	0	0	0	0	0
vsaRNA	0	1	0	1	0	0	0	0	1	1	1	1
mRNA4kb(N)	0	1	0	1	0	0	0	0	1	1	1	1
mRNA2kb(N)	0	1	0	1	0	0	0	0	1	1	1	1
mRNA2kb(C)	0	0	0	0	0	0	0	0	1	1	1	1
mRNA4kb(C)	0	0	0	0	0	0	0	0	1	1	1	1
mRNA9kb(C)	0	0	0	0	0	0	0	0	1	1	1	1
asRNA	0	0	0	0	0	0	0	0	0	0	0	0
Tat	0	0	0	0	0	0	0	0	1	1	1	1
Rev	0	0	0	0	0	0	0	0	1	1	1	1
Nef	0	0	0	0	0	0	0	0	1	1	1	1
Vpr	0	0	0	0	0	0	0	0	1	1	1	1
p24Gag	0	0	0	0	0	0	0	0	1	1	1	1
Classification	Latency attractors*									Activation attractors		

**All attractors in which p24Gag was inactive are classified as latency attractors.*

reducing 10-fold the splicing rate of nuclear mRNA of 4 kb (s_1). The nucleosomal condensation was modeled by providing square pulses of I_{HMTs}, and T-cell activation was modeled by increasing the value of the NF-κB activity rate (k_1).

The temporal effects of treatments with histone deacetylase inhibitors (HDACis), PKC agonists, P-TEFb releasers, histone methyltransferase inhibitors (HMTis), and antagonist micro-RNAs (antagomirs) on the GRN dynamics were simulated as follows: to simulate the rise on acetylation due to HDACis, we increased two-fold the reference value of the parameters of HATs activity (k_5). The increase the NF-κB levels due to PKC agonists (Mehla et al., 2010), was modeled by increasing the value of NF-κB levels (k_1) of the ODEs system. Considering that P-TEFb releasers, such as the compound JQ1, enhance the function of Tat to sequester P-TEFb and activate provirus (Li et al., 2013), we modeled this type of LRA by increasing the parameter associated to Tat activity (α_5). The effects of HMTis and antagomirs were modeled by reducing two-fold the reference value of the parameters of HMTs activity (k_8), synthesis of vsiRNA (α_2), and asRNA (α_4).

Mutant proviruses treated with HDACis were simulated by a two-fold increase in the value of the parameter of HATs activity (k_5) as a pharmacological overstimulation and setting to zero the values of the parameters for synthesis of Tat (α_5 and α_6), Nef (α_8), and Vpr (α_9 and α_{10}) as gene knockouts. The inhibition of vncRNAs was simulated by reducing 0-, 2-, 20-, and 200-fold the value of the parameters for the synthesis of vsiRNA (α_2) and asRNA (α_4). All parameters cited in this paragraph are listed in **Supplementary Material**.

Analogously to Equation (3), we define E_0 and E_j as the normalized concentration of p24Gag mRNA for the wildtype network and when σ_j is perturbed, respectively. The difference

$$\Delta E = E_j - E_0, \tag{10}$$

is a measure of the effect on the viral activation of perturbing the node σ_j in response to pharmacological treatments.

Validation of the Models

The discrete and continuous models compatibility to reproduce the behavior of HIV-1 GRN was qualitatively evaluated by comparing the dynamical states of each model to the *in vitro* dynamics of provirus genic expression. To perform this, it was calculated the attractors of the discrete model and the equilibrium points of the continuous model for the wild type GRN and mutated networks $p5'LTR\,(t) = 0$, $Tat\,(t) = 0$, $Vpr\,(t) = 0$, $Rev\,(t) = 0$, $Nef\,(t) = 0$, and $mRNA4kbN\,(t) = 0$. Then, the attractors and the equilibrium points were classified in *activation state* or *latency state* according to their p24Gag expression level (i.e., *latency state* was assigned to attractors and equilibrium points that do not express p24Gag and *activation state* was assigned when p24Gag is expressed). These results were compared to *in vitro* observations reported for the wild type provirus, defective p′5LTR mutants (Ho et al., 2013), deleted *tat* (Verhoef and Berkhout, 1999), *vpr* (Rücker et al., 2004), *rev* and *nef* proviruses (Churchill et al., 2007), as well as deletions on the splicing sites (Purcell and Martin, 1993; **Figure 3A**). Once compatibility of the models was proved, the discrete model was qualitatively validated by comparing the size of activation state (W_{on}) of the nodes perturbations $HATs\,(t) = 1$, $Tat\,(t) = 1$, $vsaRNA\,(t) = 1$ and the combinations ($NF\kappa B\,(t) = 1$, $HATs\,(t) = 1$), ($NF\kappa B\,(t) = 1$, $HMTs\,(t) = 0$), against their *in vitro* equivalences, which are treatments with HDACis (Cillo et al., 2014), P-TEFb releasers (Li et al., 2013), the use of Antagomirs (Zhang Y. et al., 2014), combinations of Bryostatin with P-TEFb releasers (Laird et al., 2015) and combinations of Bryostatin with HMTis (Bouchat et al., 2012). In **Figure 3B** is shown the outcome of this comparison, which pointed out that the discrete model is able to predict at qualitative level changes occurred on latency reversion reported *in vitro*. The continuous model was validated by comparing the levels of genomic RNAs obtained *in silico* against *ex vivo* data (Laird et al., 2015). To perform this, it was normalized the levels of p24Gag obtained with the ODEs model for making a linear regression analysis and Pearson correlation with 5% of significance ($\alpha = 0.05$; **Figure 3C**). These analysis showed that there is a significant positive relationship between *ex vivo* and *in silico* data sets, $R^2 = 0.9426$ with $p < 0.05$, which suggest that the ODEs model is able to predict variations over concentration levels of molecular components of the HIV-1 GRN.

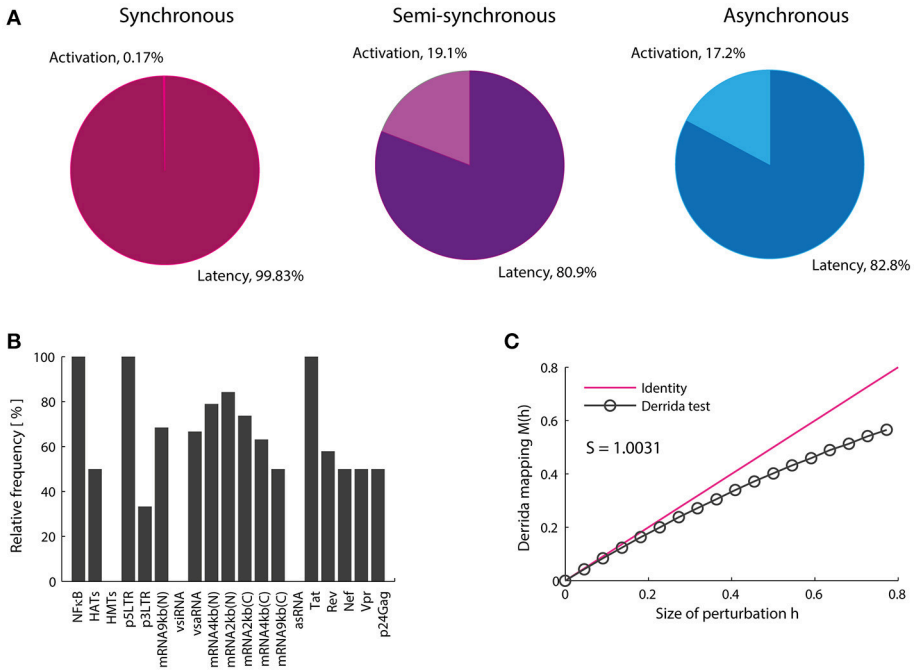


FIGURE 4 | Proviral activation is repressed by vncRNAs and rescued by Tat. **(A)** Relative weight of the activation state (in percentage). In this panel is shown the result of calculating the probability of activation using the synchronous, semi-synchronous, and asynchronous update schemes. In all cases it is shown that latency is favored over activation once provirus is integrated in the host genome. **(B)** Trajectory of activation obtained from intersection of the three update schemes. This panel shows that the presence of NF-kB, p5′LTR, Tat as well as the absence of vncRNAs (i.e., vsiRNA and asRNA) and HMTs are obligatory conditions to activate latent proviruses. **(C)** Sensitivity of the network calculated with Derrida's mapping test. The data obtained from the Boolean model suggested that Tat and vncRNAs are the main proviral regulators of latency and activation.

RESULTS

T-Cell Activation May Not Induce Expression of the Provirus

Razooky and coworkers found evidence suggesting that proviral latency is mainly regulated by the transactivation of 5′LTR mediated by Tat instead of T-cell activation, which implies that latency regulation may be an autonomous process (Razooky et al., 2015). It is in the light of this finding that, the role of epigenetic factors on the performance of Tat's autonomous behavior was investigated. To accomplish this, we analyzed the attractors of the Boolean and its basins of attraction in presence of cellular signals that stimulate epigenetic regulators such as HMTs and HATs, and activators of the NF-κB pathway like TNF. In the three update schemes it was found 12 punctual attractors (the same in the three schemes) which were classified in two groups according to expression of viral proteins as follows: (1) attractors that produce late proteins like p24Gag (*activation attractors*); and (2) attractors that lack protein expression (*latency attractors*; see **Table 3**).

The Boolean model shows that the activation attractors can be reached with or without cellular stimulation of HATs and TNF (**Table 3**), which agrees with previous observations that demonstrate the persistence of provirus expression in resting CD4+ T-cells (Razooky et al., 2015). However, this dynamics always requires the absence of the silencing produced by the HMTs activity (**Table 3**). The probability with which the provirus reaches latency and activation was investigated by calculating the relative size of the activation state (W_{on}) as well as the relative size of the latency state (W_{off}). It was found that W_{on} is always smaller than W_{off} (**Figure 4A**) even when transcription stimulatory signals like HATs and the NF-κB pathway are turned on. These results suggest that even in the context of T-cell activation, provirus may remain latent because of its autonomous dynamics, which is limited by epigenetic silencing.

Viral Non-coding RNAs Are Essential to Regulate Latency

Previous reports showed the importance of Tat as the unique virus-encoded regulator of HIV-1 autonomous behavior (Weinberger et al., 2005; Razooky et al., 2015). However, a virus-encoded siRNA that also promotes provirus activation has been found recently (Zhang Y. et al., 2014). Additionally, other virus-encoded regulators, such as vncRNAs that directly repress provirus gene expression have been found (Groen and Morris, 2013; Saayman et al., 2014; Suzuki et al., 2015). Therefore, the role of vncRNAs on the regulation of proviral latency was investigated by searching for common states in all basins of attraction of activator attractors obtained with the three updating schemes (**Figure 4B**). Using this procedure we found a set of GRN states that abrogate latency (**Figure 4B**). This set of states indicates a general pattern that results in provirus activation, which

agrees with previous reports and requires: high levels of NF-κB (Westendorp et al., 1995), no epigenetic silencing by HMTs (Jordan et al., 2003; du Chéné et al., 2007), genomic integrity of provirus (Ho et al., 2013), high levels of Tat (Weinberger et al., 2005; Razooky et al., 2015), and the absence of repressive vncRNAs (denoted by asRNA and vsiRNA; **Figure 4B**). This result suggests that Tat and repressive vncRNAs are essential virus-encoded regulators of latency establishment and activation.

HIV-1 Is Resistant to Drugs and Intracellular Perturbations

Genetic networks of organisms are able to maintain and adapt their operation in response to environmental changes. Previous studies have shown that the coexistence of robustness and adaptability observed in genetic networks is characteristic of systems operating at the critical point, i.e., at the border of chaos and order (Balleza et al., 2008). This dynamical feature has been reported for genetic networks of *A. thaliana, D. melanogaster, S. cerevisiae, E. coli, B. subtilis* (Balleza et al., 2008) as well as of mice macrophages (Nykter et al., 2008). It has been suggested that criticality is essential to ensure the evolution of any organism (Balleza et al., 2008). We investigated the presence of critical dynamics in the HIV-1 GRN. To do this, the effect of massive perturbations on the GRN was evaluated using the Derrida mapping test. When the network sensitivity S for the provirus GRN was computed, it was obtained $S = 1.0031$ which means that the network operates in a critical regime (**Figure 4C**). Therefore, this network shows equilibrium between robustness and adaptability in resting CD4+ T-cells (**Figure 4B**). This result suggests that the regulation of the expression of the HIV genome is robust against intracellular perturbations and it can be adapted in response to chronic perturbations, such as those produced during cART or treatments with LRAs. It should be noted that the HIV-1 network has constructed taking into account the activating and inhibitory interactions reported in the literature without considering criticality as a relevant criterion. The result showing that the dynamics of the HIV-1 GRN is so close to criticality is unexpected.

The Architecture of the HIV-1 GRN Allows Viral Rebounds and Persistence

Previous observations on the dynamics of Tat's positive feedback loop demonstrated that this circuit is able to amplify transcriptional fluctuations of provirus by itself, and its activity tends to decay toward a latency stable state (Weinberger et al., 2005). It has been proposed that delays on Tat's activity facilitate latency establishment (Weinberger et al., 2005), which could maintain proviral reservoirs during cART (Rouzine et al., 2015). However, it is unknown whether other viral components like Vpr, Nef, and vncRNAs modify the dynamics of Tat's circuit. In this direction, we extended previous findings by analyzing the provirus gene expression dynamics in the presence of Tat and other viral interactions that regulate proviral transcription, such as those mediated by vncRNAs and positive feedback loops of Nef and Vpr (Varin et al., 2003; Liu et al., 2014).

TABLE 4 | Equilibrium points of the HIV ODEs model.

Nodes	Latency equilibrium (arbitrary units)	Activation equilibrium (arbitrary units)
NF-κB	0	0.6243
HATs	0	0.2747
HMTs	0	0.1963
p5′LTR	0	0.3379
p3′LTR	0	0.0831
RNA9kb(N)	0	1.0559
vsiRNA	0	0.1056
vsaRNA	0	0.0211
mRNA4kb(N)	0	0.0807
mRNA2kb(N)	0	0.0279
mRNA2kb(C)	0	0.2792
mRNA4kb(C)	0	0.0154
mRNA9kb(C)	0	0.0563
asRNA	0	0.4070
Tat	0	0.1893
Rev	0	0.4035
Nef	0	1.6142
Vpr	0	0.1893
p24Gag	0	5.5836

To this end, it was used the continuous model to analyze temporal variations of the dynamics of the levels of provirus proteins and RNAs. It was performed the stability analysis of the ODEs model with a set of reference parameters (**Supplementary Material**), and found two equilibrium points that correspond to *activation* and *latency* states, i.e., the levels of p24Gag were zero for latency state and distinct to zero for activation state (**Table 4**). In this regard, the stability analysis showed that the activation state was stable and the latency state was unstable (**Figure 5A**). Then, the sensitivity of the system against fluctuations was evaluated by performing a global sensitivity analysis finding that the dynamics of the system was robust against perturbations (**Supplementary Figure 1**), and the mean value of p24Gag during activation state was 11.2 with a variance of 7.2. This suggests that once activation state is reached, the provirus expression is resistant to variations of the intracellular environment.

We searched for parameters that change stability of the equilibrium points of the GRN performing bifurcation analysis. Indeed, it was found a transcritical bifurcation (**Figure 5B**) on the value of NF-κB activation constant (k_1), parameters related to splicing of viral mRNAs (s_1), and the activity of the 5′LTR promoter (k_b). Bifurcation analysis showed that latency is stabilized when the values of these parameters are decreased (**Figure 5C**). This observation is congruent with *in vitro* reports of conditions that stabilize latency, such as low levels of NF-κB (Westendorp et al., 1995), deficient splicing sites (Purcell and Martin, 1993), and deletions on 5′LTR promoter (Dar et al., 2014) (**Table 1**). Then, we investigated the possible function of this bifurcation in the context of intracellular infection of HIV-1. To implement this, it was compared the performance of WT

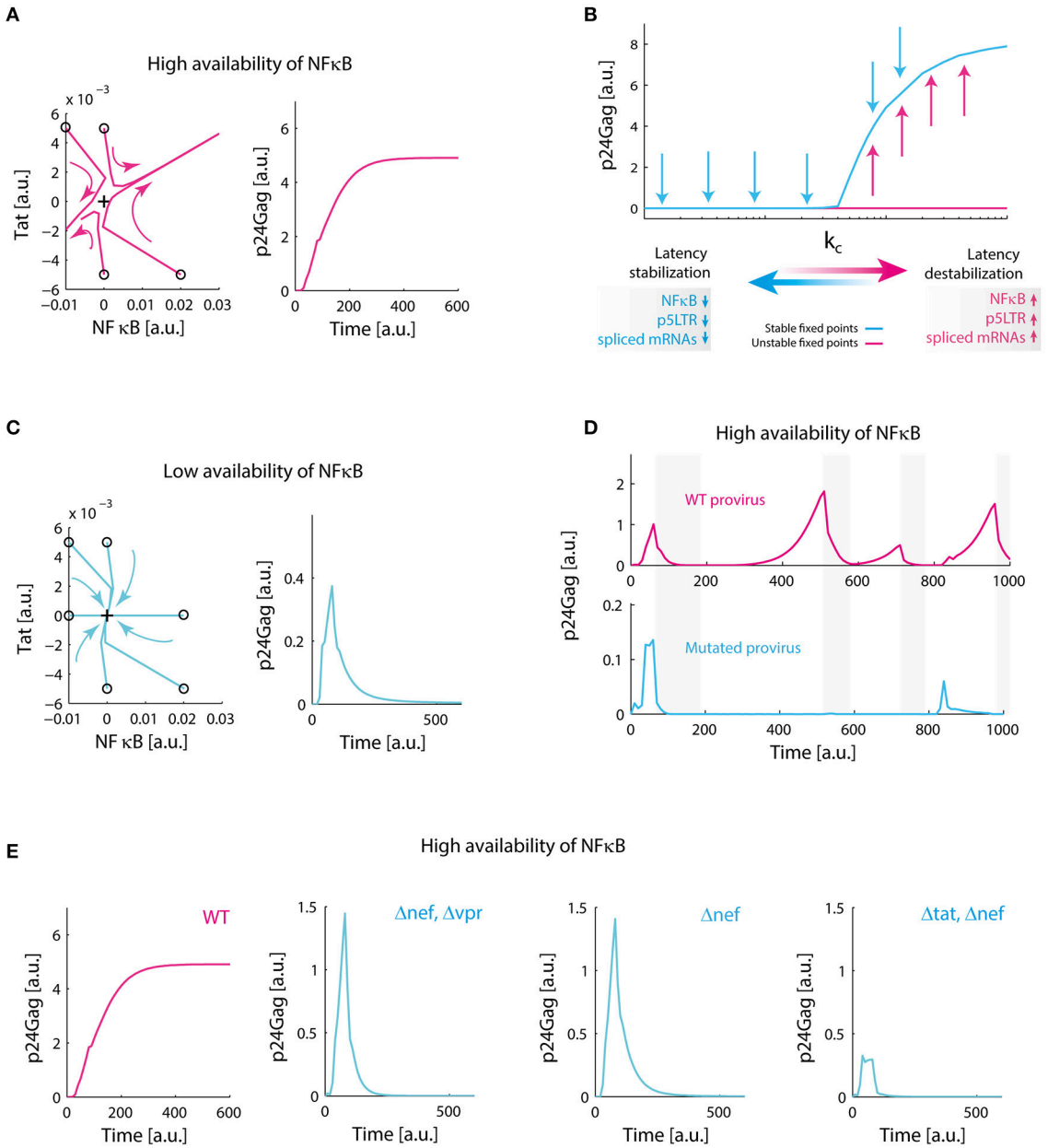


FIGURE 5 | Redundant positive feedback loops of Tat, Nef, and Vpr promote viral persistence. **(A)** Destabilization of latency in the presence of high levels of NF-κB. Phase portrait of the system around the equilibrium point corresponding to latency (black cross), and the temporal performance of the system are shown. In the phase portrait, trajectories of the system are repelled to activation state, on the temporal plot of the system; p24Gag reaches an expression stable state. This simulation was made with our reference value for NF-κB availability (k_1). **(B)** Transcritical bifurcation on the GRN. We found that variations on parameters related to availability of NF-κB, the activity of 5′LTR promoter and the splicing of viral mRNAs change the dynamical behavior of the system. The critical parameters to obtain this bifurcation are included in **Supplementary Table 5**. **(C)** Stabilization of latency. When we decreased 10-fold NF-κB availability, all trajectories in the phase portrait of the system converge to latency state (black cross), in the temporal plot this can be observed as a transient activation of protein expression that eventually decays. **(D)** Biological role of transcritical bifurcation. In the absence of this bifurcation, defective proviruses decrease their ability to relapse after a period of repression. This simulation was made by decreasing 10-fold the splicing rate of nuclear mRNA of 4 kb (s_1); gray bars indicate nucleosome compaction due to HMTs activity. **(E)** Molecular origin of transcritical bifurcation. Individually, positive feedback loops of Tat, Nef, and Vpr have a transient activity (as observed in panel **C**), however, transcritical bifurcation emerges when loops are combined. Δ*nef*, Δ*vpr,* and Δ*tat* were simulated by setting to zero the synthesis parameters of Tat, Nef, and Vpr (**Supplementary Material**). Collectively, these data suggest that redundant activation of NF-kB mediated by Tat, Nef, and Vpr ensures proviral reactivation after a period of repression.

provirus vs. mutated provirus that have attenuated splicing rates (10-fold lower of the reference value for s_1) in presence or absence of epigenetic silencing (i.e., when HMTs are active). It was found that transcritical bifurcation allows viral rebounds of the WT provirus after cellular inhibition (**Figure 5D**), which suggests that persistence may be "hardwired" on the HIV-1 genome.

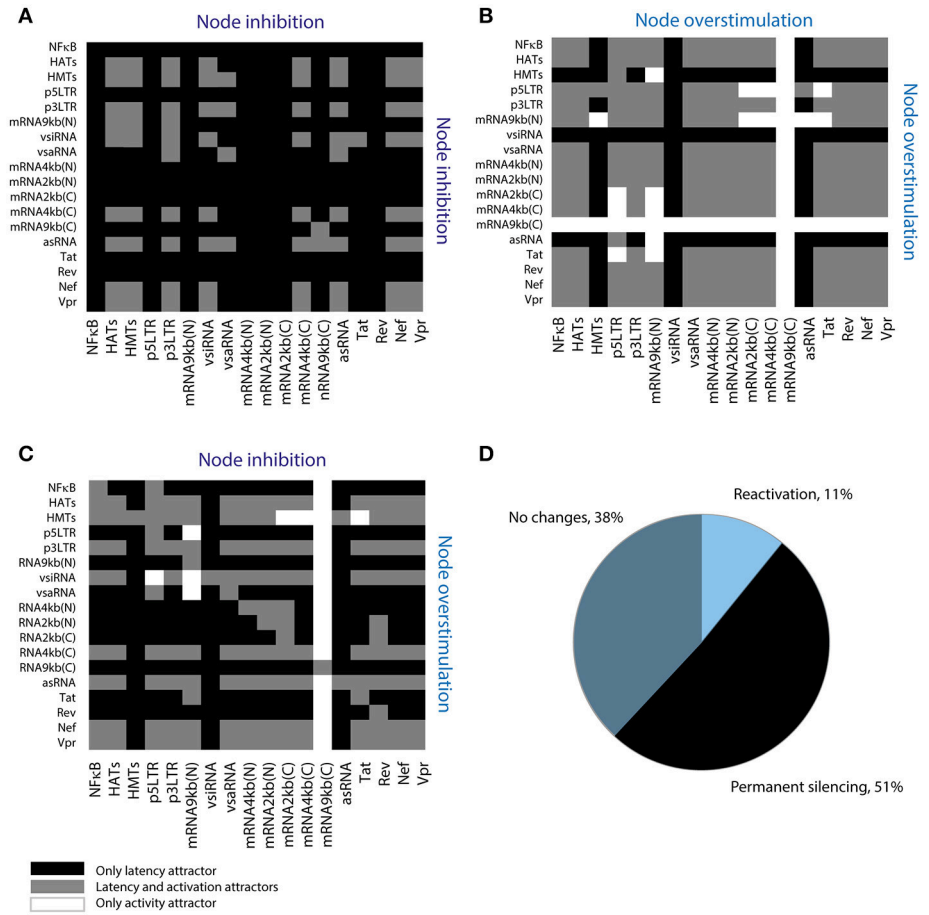


FIGURE 6 | Screening assay for reactivating perturbations. **(A)** Simultaneous inhibition of two nodes of the network. **(B)** Simultaneous overstimulation of two nodes of the network. **(C)** Inhibition and overstimulation of two nodes of the network. **(D)** Summary of screening results. This assay shows that 51% of the perturbations permanently silence provirus' expression; where "reactivation" refers to perturbations that suppress latency attractors, "no changes" refers to perturbations that allow the coexistence of latency and activation attractors, and "permanent silencing" refers to perturbations that abrogate activation attractors.

On the other hand, proviruses that lack transcritical bifurcation can be easily controlled by the host's HMTs (**Figure 5D**). These results suggest that the transcritical bifurcation of the provirus GRN may provide two dynamical behaviors: (1) for repressive transcriptional environments, such as during cART, the provirus latency will be stabilized allowing reservoirs maintenance, and (2) for non-repressive transcriptional environments, the provirus favors a strong activation in order to ensure the production of viral progeny and to counteract the intracellular silencing mechanisms. These properties may explain the viral rebounds after cART and why HIV-1 cannot be silenced by host.

The Activating Core of the GRN Consists of the Positive Feedback Loops of Tat, Nef, and Vpr

The molecular basis of the transcritical bifurcation was investigated comparing the activity of intact provirus vs. the activity of mutant proviruses. Mutant proviruses were simulated in the continuous model by setting to zero all parameters related to the synthesis of viral proteins Tat, Nef, and Vpr. It was observed that the Tat's positive feedback circuit always produces a stable branch on latency state, which in biological terms is a transient activation followed by latency stabilization dynamics as reported by Weinberger et al. (2005) (**Figure 5E**). However, combining Tat positive feedback with Vpr and Nef produces the transcritical bifurcation, in which latency can be destabilized (**Figure 5B**). We also observed that in the absence of Tat the remaining positive feedback loops were able to temporarily perturb latency during stimulation, producing transitory gene activation, but their effect was negligible compared to that observed in the presence of Tat (**Figure 5E**). Thus, the transcritical bifurcation is sustained by all the positive feedback loops of the viral proteins Tat, Vpr, and Nef (**Figure 5E**). Considering that all the positive feedback loops of HIV-1 promote NF-κB activation (**Figure 1**), it is reasonable to think that the redundancy on NF-κB stimulation is the cause of the transcritical bifurcation and its amplifying properties.

Permanent Stabilization of Latency Occurs More Frequently Than Reactivation

Recently, it has been proposed that compounds that increase fluctuations of transcriptional basal levels may enhance the performance of LRAs (Dar et al., 2014). Such compounds indirectly target the 5′LTR promoter, increasing its activity. We extended this result by searching for sensitive interactions that could increase proviral reactivation in the presence of LRAs. To this end, it was used the Boolean model to explore all possible perturbations of the provirus GRN by combining inhibition and stimulation of the GRN nodes using a screening assay (**Figure 6**). It was found that 51% of the perturbations eliminated activation attractors, which suggests that those perturbations are able to induce permanent silencing of the provirus (**Figure 6D**). On the other hand, it was found that only 28 of the 648 theoretical perturbations can be performed *in vivo* using current LRAs and antagomirs (**Table 5**). Remarkably, some of these perturbations have not been tested yet. These results suggest that it would be easier to induce the permanent silencing of HIV-1 proviruses rather than reactivating them (**Figure 6D**).

Inhibition of HMTs and Stimulation of P-TEFb Increases Proviral Reactivation

We then characterized the dynamical properties of 28 promising perturbations produced with LRAs and antagomirs (**Table 5**). To do this, the dynamical performance of each perturbation was compared to the dynamics of the WT provirus. It was used the Boolean model to calculate the relative size of the activation state (W_{on}) and the difference of sensitivity (ΔS). Similarly, it was used the ODEs model to determine the difference of p24Gag expression (ΔE) for each perturbation. It was found that all reactivation perturbations increased W_{on}, except HATs(+) (**Figure 7A**) which is equivalent to using HDACis (**Table 5**). Moreover, all reactivating perturbations decreased network sensitivity (**Figure 7B**) and the ODEs model showed that all perturbations, except HATs(+), increased the expression of p24Gag (**Figure 7C**). Remarkably, the discrete model showed that inhibition of HMTs and overstimulation of Tat, i.e., HMTs(−), Tat(+) precludes latency attractors, which means that provirus is always active (**Figure 7A**). Analogously, the ODEs model showed that HMTs(−), Tat(+) increases ΔE to the maximum (**Figure 7C**). It is important to note that the pharmacological equivalence of HMTs(−), Tat(+) can be implemented with HMTis and P-TEFb releasers (Li et al., 2013; **Table 5**). In **Table 5** are shown the pharmacological treatment equivalent for the other latency reversing perturbations.

The Performance of LRAs Is Hindered by vncRNAs

Recent reports showed that HDACis are not effective to reactivate latent proviruses (Bullen et al., 2014; Cillo et al., 2014). In agreement with these reports, the models showed that HDACis do not produce changes in the activation state (**Figure 7A**) and do not increase p24Gag expression levels (**Figure 7C**). However, it has been reported that HDACis increase transcription of provirus (Mohammadi et al., 2014). To explain the HDCAis underperformance, the existence of unknown post-transcriptional mechanisms that counteract protein synthesis have been proposed (Mohammadi et al., 2014). Furthermore, it has been reported that HDACis like SAHA (Vorinostat) may increase the levels of cellular non-coding RNAs (Lee et al., 2009). Taken together these observations suggest that HDACis increase provirus transcription as well as the levels of viral and cellular non-coding RNAs, which contributes to silencing protein expression of provirus. We explored this hypothesis by comparing W_{on}, ΔS, and ΔE for each HDACis perturbation with and without vncRNAs (see section Methods). It was found that the suppression of vncRNAs enhances HDACis performance,

TABLE 5 | Proposed treatments to reverse latency and their current status.

Perturbation	Equivalent treatments	References
HMTs (–)	HMTis	Bouchat et al., 2012
HMTs (–), vsiRNA (-)	HMTis + Antagomirs	*
HMTs (–), asRNA (-)	HMTis + Antagomirs	*
vsiRNA (–)	Antagomirs	*
vsiRNA (–), asRNA (–)	Antagomirs	*
asRNA (–)	Antagomirs	Saayman et al., 2014
NF-κB (+)	PKC agonists	Mehla et al., 2010
NF-κB (+), HATs (+)	PKC agonists + HDACis	Laird et al., 2015
NF-κB (+), vsaRNA (+)	PKC agonists + Antagomirs	*
NF-κB (+), Tat (+)	PKC agonists + P-TEFb releasers	Laird et al., 2015
HATs (+)	HDACis	Bullen et al., 2014
HATs (+), vsaRNA (+)	HDACis + Antagomirs	*
HATs (+), Tat (+)	HDACis+ P-TEFb releasers	Darcis et al., 2015
vsaRNA (+)	vsaRNA	Zhang Y. et al., 2014
vsaRNA (+), Tat (+)	Antagomirs + P-TEFb releasers	*
Tat (+)	P-TEFb releasers	Darcis et al., 2015
NF-κB (+), HMTs (–)	PKC agonists + HMTis	Bouchat et al., 2012
NF-κB (+), vsiRNA (–)	PKC agonists + Antagomirs	*
NF-κB (+), asRNA (–)	PKC agonists + Antagomirs	*
HATs (+), HMTs (–)	HDACis + HMTis	Bouchat et al., 2012
HATs (+), vsiRNA (–)	HDACis + Antagomirs	*
HATs (+), asRNA (–)	HDACis + Antagomirs	*
vsaRNA (+), HMTs (–)	Antagomirs + HMTis	*
vsaRNA (+), vsiRNA (–)	Antagomirs	*
vsaRNA (+), asRNA (–)	Antagomirs	*
Tat (+), HMTs (–)	P-TEFb releasers + HMTis	*
Tat (+), vsiRNA (–)	P-TEFb releasers + Antagomirs	*
Tat (+), asRNA (–)	P-TEFb releasers + Antagomirs	*

**Not evaluated yet. The most promising pharmacological perturbations that can be performed to reactivate latent proviruses are included in the table. The corresponding treatment for each perturbation can be implemented as follows: Increasing NF-κB levels [denoted by NF-κB (+)] can be obtained using PKC agonists such as bryostatin (Mehla et al., 2010). Increasing acetylation levels of provirus [denoted by HATs (+)] can be obtained by protecting HATs dependent acetylation with inhibitors of histone deacetylases (HDACis) such as romidepsin or Suberoylanilide Hydroxamic Acid acid (SAHA, Vorinostat) (Reuse et al., 2009; Bullen et al., 2014). Increasing transcriptional effects of Tat [denoted by Tat (+)] can be induced with P-TEFb releasers like JQ1 (Li et al., 2013). Suppression of HMTs activity [denoted by HMTs(–)] can be performed with inhibitors of those enzymes (HMTis) such as chaetocin (Bouchat et al., 2012). Inhibition of vncRNAs, like asRNA and vsiRNA [denoted by asRNA (–) and vsiRNA (–)] can be performed using antagomirs (Yeung et al., 2009; Saayman et al., 2014).*

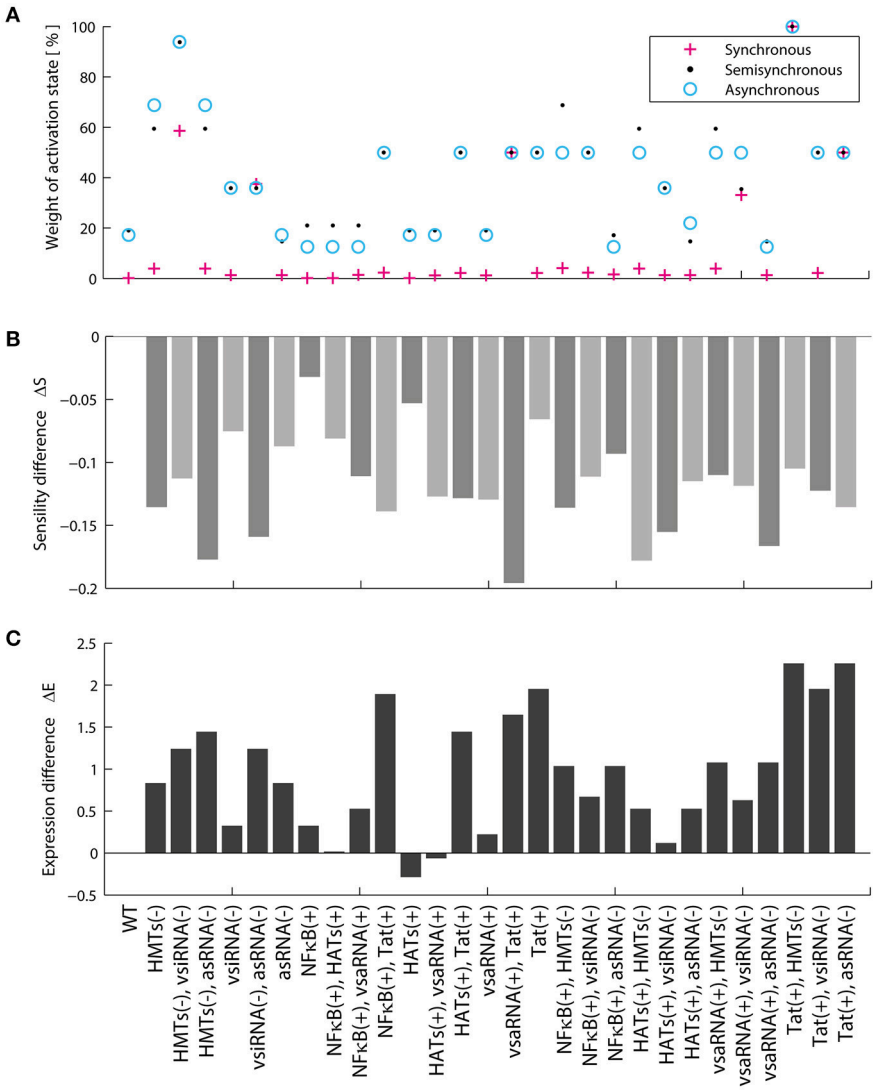


FIGURE 7 | Dynamical features of activating perturbations with LRAs. **(A)** Relative weight of activation state for each activating perturbation. **(B)** Sensitivity difference for each activating perturbation. **(C)** Difference of p24Gag expression for each activating perturbation. In general, all LRAs perturbations increase the weight of the activation state and protein expression.

increasing the values of W_{on} (**Figure 8A**), ΔS (**Figure 8B**), and the expression levels of p24Gag (**Figure 8C**). These data suggest that HDACis may promote the synthesis of vncRNAs, which may explain why these LRAs increase provirus transcription but not protein expression (**Figure 8D**).

Inhibition of vncRNAs Is Not Sufficient to Stimulate Proviral Reactivation

The results just presented indicate that inhibiting vncRNAs could enhance the effect of LRAs (**Figure 8D**). However, it is not clear whether vncRNAs inhibition can also stimulate the reactivation of mutant proviruses. Therefore, we used the ODEs model to address this question and compared the expression levels of p24Gag in defective provirus treated with HDACis at different intensities of vncRNAs inhibition. It was found that mutant proviruses that lack the Tat protein can be reactivated to a lesser extent than intact proviruses (**Supplementary Figure 2**). However, defective proviruses that lack two or more positive feedback loops cannot be reactivated, even with the inhibition of vncRNAs (**Supplementary Figure 2**). These results suggest that inhibition of vncRNAs cannot ensure the total reactivation of proviral reservoirs.

DISCUSSION AND CONCLUDING REMARKS

The long-lived latent reservoirs of HIV-1 are the main barrier to eradicate it. Several efforts to purge viral reservoirs have

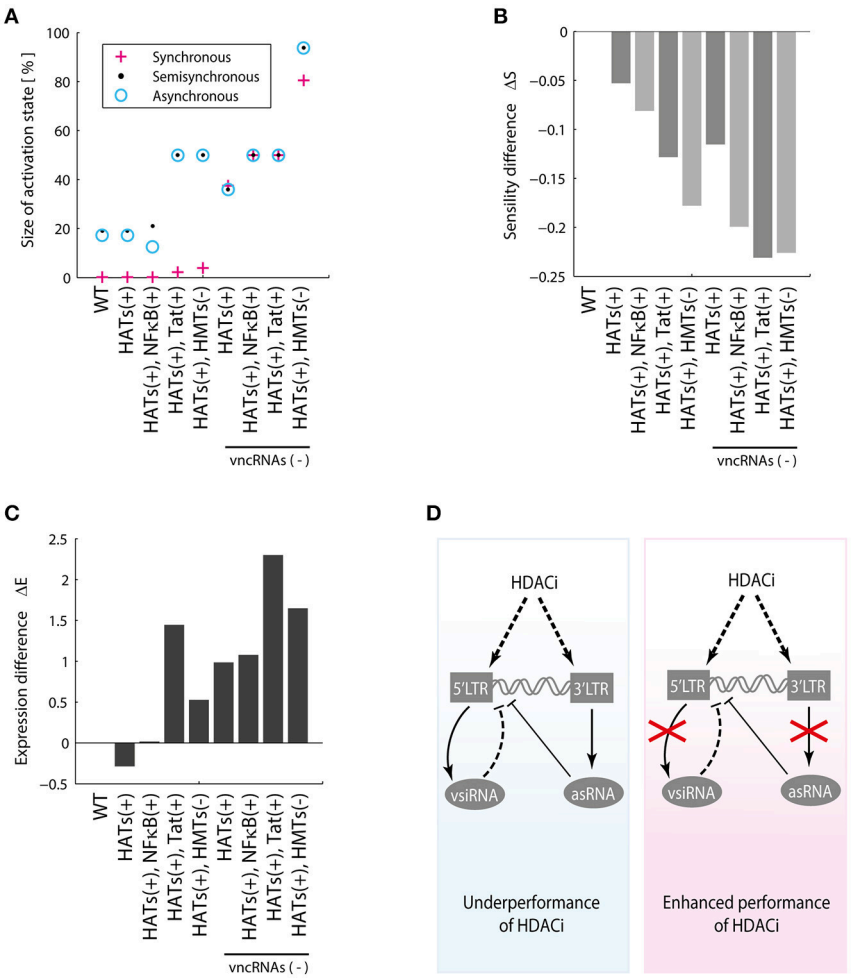


FIGURE 8 | HDACis indirectly increase vncRNAs. **(A)** Relative weight of activation state, **(B)** Sensitivity difference, **(C)** Difference of p24Gag expression with and without vncRNAs, denoted by vncRNAs (–). **(D)** These data suggest that LRAs like HDACis indirectly increase the synthesis of vncRNAs, which hinders their reactivating effects. The suppression of the vncRNAs may enhance the effectiveness of HDACis.

been performed using LRAs, unfortunately none of them were effective *in vivo* (Bullen et al., 2014). Until now it is not known the causes of the underperformance of LRAs. In this work, we analyzed *in silico* the functioning of the provirus' gene expression in order to investigate the ineffectiveness of LRAs. To this end, we constructed the GRN of provirus and modeled its dynamics using ODEs and logic rules. Both models predicted that vncRNAs are the main negative regulators of the gene expression of provirus and they are also implicated in the underperformance of LRAs. Finally, both models predicted that treatments with HMTis and P-TEFb releasers are the best way to maximize latency reversion.

Traditionally it has been thought that Tat is the only virus-encoded regulator of the HIV latency. However, recent evidence shows that vncRNAs are also essential to control proviral latency. Saayman and colleagues characterized an HIV-encoded long anti-sense RNA which its inhibition triggers reactivation in latently infected cells (Saayman et al., 2014). Zapata and coworkers showed that this long anti-sense RNA is able to silence the gene expression of provirus by stimulating HMTs (Zapata et al., 2017). Thus, we investigated the role of vncRNAs on the dynamics of provirus' gene expression. The first dynamical particularity of the GRN was that the weight of the latency state (W_{off}) was higher than the weight of the activation state (W_{on}), regardless the cell's activation state (**Figure 4A**). After analyzing the set of intracellular environments that activate the GRN (**Figure 4B**), we noted that activation requires the presence of Tat and the absence of vncRNAs. Additionally, the inhibition of vncRNAs increased the W_{on} (**Figure 8A**). Taken together these results indicate that vncRNAs are the main negative regulators of the provirus' genic expression.

The next question to address was how vncRNAs and Tat operate together to regulate latency. Previous reports demonstrate that Tat's positive feedback loop has a strong transient activation that eventually decays to a stable latency state (Weinberger et al., 2005; Weinberger and Shenk, 2007). The same behavior was observed on the Tat's circuit of the

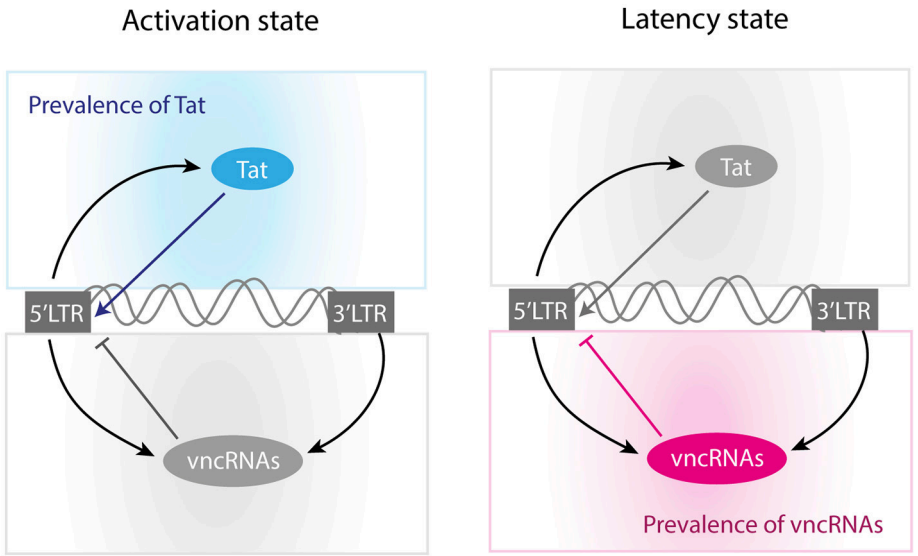


FIGURE 9 | Molecular mechanisms of self-regulation of proviral latency. According to our results, activation state is mainly produced when Tat concentration reaches high levels. On the other hand, the provirus induces its latency when Tat's concentration is not optimal and the levels of vncRNAs are high.

GRN (**Figure 5E**), as well as in other positive feedback loops mediated by Nef and Vpr (**Figure 5E**). Interestingly, we found that a transcritical bifurcation appears when these circuits were combined (**Figure 5B**), and such a bifurcation allows gene expression rebounds after long periods of repression (**Figure 5D**). It seems likely that the Tat's circuit is enhanced by Nef and Vpr in order to overcome the downregulation of vncRNAs and the host. However, an uncontrolled enhancement of the gene expression of provirus could have negative effects on the viral reservoirs. Rouzine and colleagues found that a high rate of proviral activation avoids the establishment of latent reservoirs, which decreases the prevalence of HIV-1 (Rouzine et al., 2015). They also observed that fluctuations on the transient activity of Tat, decreases the frequency of provirus' activation which stabilizes viral reservoirs (Rouzine et al., 2015). Expanding these observations, our results showed that in addition to Tat's fluctuations, vncRNAs also reduce the activation of provirus. Thus, vncRNAs together with Tat's transient activity may be responsible for the chronic stabilization of latency, condition required to maintain the viral reservoirs (**Figure 9**).

Furthermore, we investigated the role of vncRNAs on the underperformance of LRAs. The screening assay (**Figure 6**) showed that 28 perturbations of the GRN can be implemented with LRAs and antagomirs (**Table 5**), being the combination of HMTis with P-TEFb releasers the most prominent of all. However, perturbations made with HDACis did not increase protein expression of provirus (**Figure 7**), as reported by Cillo et al. (2014). Mohammadi et al found that HDACis only increase provirus' transcription but did not affect protein expression (Mohammadi et al., 2014). They proposed that this occurs because of post-transcriptional mechanisms that hinder protein expression (Mohammadi et al., 2014). In this direction, our results predicted that the levels of vncRNAs increased in response to HDACis (**Figure 8**). Hence, it seems likely that treatments with HDACis stimulate proviral transcription as well as vncRNAs, which eventually avoids protein expression. This hypothesis may explain the underperformance of treatments with LRAs reported *in vivo*.

The final question to address was how to enhance the performance of LRAs. The screening assay showed 28 feasible treatments to disrupt latency by using micro-RNAs and current LRAs (**Table 5**). In this direction the treatment that maximizes the probability to reactivate proviruses (given by the value of W_{on}) uses HMTis and P-TEFb releasers (**Figure 7A**). The action mechanism of this treatment consists in increasing Tat's levels with P-TEFb releasers while the activity of HMTs is blocked, which is the main downstream target of vncRNAs (Zapata et al., 2017). Therefore, blocking molecular effectors of vncRNAs and enhancing Tat activity is the best way to increase viral reactivation. It is of our interest to test the effectiveness of the treatments proposed in **Table 5** with *ex vivo* cultures obtained from HIV patients, in order to determine whether such treatments could be promising for therapeutic implementation.

Nevertheless, our results also showed an interesting scenario that has a distinct approach to control HIV-1. The screening assay showed that 51% of perturbations permanently silence the provirus genic expression (**Figure 6D**). It is noteworthy to say that the most of perturbations that permanently silence the provirus, inhibit nodes related to proviral transcription such as p5′LTR and unspliced, spliced and partially spliced viral mRNAs (**Figure 6A**). This implicates that HIV-1 can be permanently controlled by the induction of hypermutation of its genome. A possible mechanism to implement this strategy can be achieved with APOBEC3G, which is the enzyme that naturally hypermutates HIV-1 as a part of intracellular antiviral response. In this context, APOBEC3G is inhibited by Vif in order to allow

the progression of HIV-1 infection. However, recent findings suggest that drugs that stimulates ASK1 (apoptosis signal-regulating kinase 1) also restore the APOBEC3G function even in presence of Vif (Miyakawa et al., 2015). Thus, an alternative path to control HIV-1 infection may employ APOBEC3G inducers in conjunction with cART.

Current treatments to reactivate latent proviruses may fail because HIV uses its vncRNAs as negative regulators to maintain latency. Some LRAs like HDACis could increase the levels of vncRNAs, consequently reducing their effectiveness to revert latently infected cells. Our results suggest that the best treatment to avoid the repressive effects of vncRNAs is to use an HMTis like chaetocin, together with P-TEFb enhancers. Treatment that could have potential for efficient reactivation of the HIV-1 provirus should be clinically tested.

AUTHOR CONTRIBUTIONS

AB and CT-S conceptualized, designed, and performed all computational experiments of this study. RG and JD contributed designing experiments, interpreting, and supervising this study. AB and CT-S wrote the first draft of the manuscript. RG and JD reviewed drafts and approved final version of the manuscript.

FUNDING

This work was supported by CONACYT, and by the PRODEP funding program of the Universidad Autónoma del Estado de Morelos. AB was supported by CONACYT through the Ph.D. Scholarship number 27667. CT-S was supported by a PRODEP Postdoctoral grant 103.5/14/11795.

ACKNOWLEDGMENTS

We thank Dr. Maximino Aldana for his constructive comments and suggestions. We also thank Erika Juarez Luna for logistical support.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphys.2018.01364/full#supplementary-material

REFERENCES

Aldana, M. (2003). Boolean dynamics of networks with scale-free topology. *Phys. D* 185, 45–66. doi: 10.1016/S0167-2789(03)00174-X

Balleza, E., Alvarez-Buylla, E. R., Chaos, A., Kauffman, S., Shmulevich, I., and Aldana, M. (2008). Critical dynamics in genetic regulatory networks: examples from four kingdoms. *PLoS ONE* 3:e2456. doi: 10.1371/journal.pone.0002456

Bouchat, S., Gatot, J.-S., Kabeya, K., Cardona, C., Colin, L., Herbein, G., et al. (2012). Histone methyltransferase inhibitors induce HIV-1 recovery in resting CD4(+) T cells from HIV-1-infected HAART-treated patients. *AIDS* 26, 1473–1482. doi: 10.1097/QAD.0b013e32835535f5

Bullen, C. K., Laird, G. M., Durand, C. M., Siliciano, J. D., and Siliciano, R. F. (2014). New *ex vivo* approaches distinguish effective and ineffective single agents for reversing HIV-1 latency *in vivo*. *Nat. Med*. 20, 425–429. doi: 10.1038/nm.3489

Churchill, M. J., Chiavaroli, L., Wesselingh, S. L., and Gorry, P. R. (2007). Persistence of attenuated HIV-1 Rev alleles in an epidemiologically linked cohort of long-term survivors infected with nef-deleted virus. *Retrovirology* 4:43. doi: 10.1186/1742-4690-4-43

Cillo, A. R., Sobolewski, M. D., Bosch, R. J., Fyne, E., Piatak, M., Coffin, J. M., et al. (2014). Quantification of HIV-1 latency reversal in resting CD4+ T Cells from patients on suppressive antiretroviral therapy. *Proc. Natl. Acad. Sci. U.S.A*. 111, 7078–7083. doi: 10.1073/pnas.1402873111

Cohn, L. B., Silva, I. T., Oliveira, T. Y., Rosales, R. A., Parrish, E. H., Learn, G. H., et al. (2015). HIV-1 integration landscape during latent and active infection. *Cell* 160, 420–432. doi: 10.1016/j.cell.2015.01.020

Dar, R. D., Hosmane, N. N., Arkin, M. R., Siliciano, R. F., and Weinberger, L. S. (2014). Screening for noise in gene expression identifies drug synergies. *Science* 344, 1392–1396. doi: 10.1126/science.1250220

Darcis, G., Kula, A., Bouchat, S., Fujinaga, K., Corazza, F., Ait-Ammar, A., et al. (2015). An in-depth comparison of latency-reversing agent combinations in various *in vitro* and *ex vivo* HIV-1 latency models identified bryostatin-1+JQ1 and ingenol-B+JQ1 to potently reactivate viral gene expression. *PLoS Pathogens* 11:e1005063. doi: 10.1371/journal.ppat.1005063

Deeks, S. G. (2012). HIV: shock and kill. *Nature* 487, 439–440. doi: 10.1038/487439a

Derrida, B., and Pomeau, Y. (1986). Random networks of automata: a simple annealed approximation. *Europhys. Lett*. 1, 45–49. doi: 10.1209/0295-5075/1/2/001

Dinoso, J. B., Kim, S. Y., Wiegand, A. M., Palmer, S. E., Gange, S. J., Cranmer, L., et al. (2009). Treatment intensification does not reduce residual HIV-1 viremia in patients on highly active antiretroviral therapy. *Proc. Natl. Acad. Sci. U.S.A*. 106, 9403–9408. doi: 10.1073/pnas.0903107106

du Chéné, I., Basyuk, E., Lin, Y. L., Triboulet, R., Knezevich, A., Chable-Bessia, C., et al. (2007). Suv39H1 and HP1γ are responsible for chromatin-mediated HIV-1 transcriptional silencing and post-integration latency. *EMBO J*. 26, 424–435. doi: 10.1038/sj.emboj.7601517

Gershenson, C. (2002). *Classification of Random Boolean Networks. Computational Complexity; Discrete Mathematics; Dynamical Systems; Cellular Automata and Lattice Gases*. Available online at: http://arxiv.org/abs/cs/0208001

Groen, J. N., and Morris, K. V. (2013). Chromatin, non-coding RNAs, and the expression of HIV. *Viruses* 5, 1633–1645. doi: 10.3390/v5071633

Hernandez-Vargas, E. A. (2017). Modeling kick-kill strategies toward HIV Cure. *Front. Immunol*. 8:995. doi: 10.3389/fimmu.2017.00995

Hill, A. L., Rosenbloom, D. I. S., Fu, F., Nowak, M. A., and Siliciano, R. F. (2014). Predicting the outcomes of treatment to eradicate the latent reservoir for HIV-1. *Proc. Natl. Acad. Sci. U.S.A*. 111, 13475–13480. doi: 10.1073/pnas.1406663111

Ho, Y. C., Shan, L., Hosmane, N. N., Wang, J., Laskey, S. B., Rosenbloom, D. I. S., et al. (2013). Replication-competent noninduced proviruses in the latent reservoir increase barrier to HIV-1 Cure. *Cell* 155, 540–551. doi: 10.1016/j.cell.2013.09.020

Jordan, A., Bisgrove, D., and Verdin, E. (2003). HIV reproducibly establishes a latent infection after acute infection of T cells *in vitro*. *EMBO J*. 22, 1868–1877. doi: 10.1093/emboj/cdg188

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol*. 22, 437–467. doi: 10.1016/0022-5193(69)90015-0

Krawitz, P., and Shmulevich, I. (2007). Basin entropy in Boolean network ensembles. *Phys. Rev. Lett*. 98:158701. doi: 10.1103/PhysRevLett.98.158701

Laird, G. M., Bullen, C. K., Rosenbloom, D. I. S., Martin, A. R., Hill, A. L., Durand, C. M., et al. (2015). *Ex vivo* analysis identifies effective HIV-1 latency–reversing drug combinations. *J. Clin. Invest*. 125, 1901–1912. doi: 10.1172/JCI80142

Lee, E. M., Shin, S., Cha, H. J., Yoon, Y., Bae, S., Jung, J. H., et al. (2009). Suberoylanilide Hydroxamic Acid (SAHA) changes microRNA expression

profiles in A549 human non-small cell lung cancer cells. *Int. J. Mol. Med*. 24, 45–50. doi: 10.3892/ijmm_00000204

Li, Z., Guo, J., Wu, Y., and Zhou, Q. (2013). The BET bromodomain inhibitor JQ1 activates HIV latency through antagonizing Brd4 inhibition of Tat-transactivation. *Nucleic Acids Res*. 41, 277–287. doi: 10.1093/nar/gks976

Liu, R., Lin, Y., Jia, R., Geng, Y., Liang, C., Tan, J., et al. (2014). HIV-1 Vpr stimulates NF-κB and AP-1 signaling by activating TAK1. *Retrovirology* 11:45. doi: 10.1186/1742-4690-11-45

Mehla, R., Bivalkar-Mehla, S., Zhang, R., Handy, I., Albrecht, H., Giri, S., et al. (2010). Bryostatin modulates latent HIV-1 infection via PKC and AMPK signaling but inhibits acute infection in a receptor independent manner. *PLoS ONE* 5:e11160. doi: 10.1371/journal.pone.0011160

Miyakawa, K., Matsunaga, S., Kanou, K., Matsuzawa, A., Morishita, R., Kudoh, A., et al.(2015). ASK1 restores the antiviral activity of APOBEC3G by disrupting HIV-1 Vif-mediated counteraction. *Nat. Commun*. 6:6945. doi: 10.1038/ncomms7945

Mohammadi, P., di Iulio, J., Muñoz, M., Martinez, R., Bartha, I., Cavassini, M., et al. (2014). Dynamics of HIV latency and reactivation in a primary CD4+ T cell model. *PLoS Pathogens* 10:e1004156. doi: 10.1371/journal.ppat.1004156

Nykter, M., Price, N. D., Aldana, M., Ramsey, S. A., Kauffman, S. A., Hood, L. E., et al. (2008). Gene expression dynamics in the macrophage exhibit criticality. *Proc. Natl. Acad. Sci. U.S.A*. 105, 1897–1900. doi: 10.1073/pnas.0711525105

Purcell, D. F., and Martin, M. A. (1993). Alternative splicing of human immunodeficiency virus type 1 mRNA modulates viral protein expression, replication, and infectivity. *J. Virol*. 67, 6365–6378.

Razooky, B. S., Pai, A., Aull, K., Rouzine, I. M., and Weinberger, L. S. (2015). A hardwired HIV latency program. *Cell* 160, 990–1001. doi: 10.1016/j.cell.2015.02.009

Reuse, S., Calao, M., Kabeya, K., Guiguen, A., Gatot, J. S., Quivy, V., et al. (2009). Synergistic activation of HIV-1 expression by deacetylase inhibitors and prostratin: implications for treatment of latent infection. *PLoS ONE* 4:e6093. doi: 10.1371/journal.pone.0006093

Romani, B., Engelbrecht, S., and Glashoff, R. H. (2010). Functions of Tat: the versatile protein of human immunodeficiency virus type 1. *J. Gen. Virol*. 91, 1–12. doi: 10.1099/vir.0.016303-0

Rouzine, I. M., Weinberger, A. D., and Weinberger, L. S. (2015). An evolutionary role for HIV latency in enhancing viral transmission. *Cell* 160, 1002–1012. doi: 10.1016/j.cell.2015.02.017

Rücker, E., Grivel, J.-C., Münch, J., Kirchhoff, F., and Margolis, L. (2004). Vpr and Vpu are important for efficient human immunodeficiency virus type 1 replication and CD4+ T-cell depletion in human lymphoid tissue *ex vivo*. *J. Virol*. 78, 12689–12693. doi: 10.1128/JVI.78.22.12689-12693.2004

Saayman, S., Ackley, A., Turner, A. W., Famiglietti, M., Bosque, A., Clemson, M., et al. (2014). An HIV-encoded antisense long noncoding RNA epigenetically regulates viral transcription. *Mol. Ther*. 22, 1164–1175. doi: 10.1038/mt.2014.29

Siliciano, J. D., Kajdas, J., Finzi, D., Quinn, T. C., Chadwick, K., Margolick, J. B., et al. (2003). Long-term follow-up studies confirm the stability of the latent Reservoir for HIV-1 in resting CD4+ T cells. *Nat. Med*. 9, 727–728. doi: 10.1038/nm880

Siliciano, R. F., and Greene, W. C. (2011). HIV latency. *Cold Spring Harb. Perspect. Med*. 1:a007096. doi: 10.1101/cshperspect.a007096

Suzuki, K., Ahlenstiel, C., Marks, K., and Kelleher, A. D. (2015). Promoter targeting RNAs: unexpected contributors to the control of HIV-1 transcription. *Mol. Ther*. 4:e222. doi: 10.1038/mtna.2014.67

Varin, A., Manna, S. K., Quivy, V., Decrion, A. Z., Van Lint, C., Herbein, G., et al.(2003). Exogenous Nef protein activates NF-κB, AP-1, and c-Jun N-terminal kinase and stimulates hiv transcription in promonocytic cells: role in AIDS pathogenesis. *J. Biol. Chem*. 278, 2219–2227. doi: 10.1074/jbc.M209622200

Verhoef, K., and Berkhout, B. (1999). A second-site mutation that restores replication of a tat-defective human immunodeficiency virus. *J. Virol*. 73, 2781–2789.

Weinberger, L. S., Burnett, J. C., Toettcher, J. E., Arkin, A. P., and Schaffer, D. V. (2005). Stochastic gene expression in a lentiviral positive-feedback Loop: HIV-1 Tat fluctuations drive phenotypic diversity. *Cell* 122, 169–182. doi: 10.1016/j.cell.2005.06.006

Weinberger, L. S., and Shenk, T. (2007). An HIV feedback resistor: auto-regulatory circuit deactivator and noise buffer. *PLoS Biol*. 5:e9. doi: 10.1371/journal.pbio.0050009

Westendorp, M. O., Shatrov, V. A., Schulze-Osthoff, K., Frank, R., Kraft, M., Los, M., et al. (1995). HIV-1 Tat potentiates TNF-induced NF-Kappa B activation and cytotoxicity by altering the cellular redox state. *EMBO J*. 14, 546–554.

Yeung, M. L., Bennasser, Y., Watashi, K., Le, S. Y., Houzet, L., and Jeang, K. T. (2009). Pyrosequencing of small non-coding RNAs in HIV-1 infected cells: evidence for the processing of a viral-cellular double-stranded RNA hybrid. *Nucleic Acids Res*. 37, 6575–6586. doi: 10.1093/nar/gkp707

Zapata, J. C., Campilongo, F., Barclay, R. A., and Demarino, C. (2017). The human immunodeficiency virus 1 ASP RNA promotes viral latency by recruiting the polycomb repressor complex 2 and promoting nucleosome assembly. *Virology* 506, 34–44. doi: 10.1016/j.virol.2017.03.002

Zhang, Q., Bhattacharya, S., Conolly, R. B., Clewell, H. J., Kaminski, N. E., and Andersen, M. E. (2014). Molecular signaling network motifs provide a mechanistic basis for cellular threshold responses. *Environ. Health Perspect*. 122, 1261–1270. doi: 10.1289/ehp.1408244

Zhang, Y., Fan, M., Geng, G., Liu, B., Huang, Z., Luo, H., et al. (2014). A novel HIV-1-encoded microRNA enhances its viral replication by targeting the TATA box region. *Retrovirology* 11:23. doi: 10.1186/1742-4690-11-23

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

ORIGINAL RESEARCH
published: 09 October 2018
doi: 10.3389/fphys.2018.01335

Evaluating Uncertainty in Signaling Networks Using Logical Modeling

Kirsten Thobe[1,2]*, *Christina Kuznia*[3,4], *Christine Sers*[3] and *Heike Siebert*[1]

[1] Group for Discrete Biomathematics, Department for Mathematics and Computer Science, Freie Universität Berlin, Berlin, Germany, [2] Group for Mathematical Modelling of Cellular Processes, Max-Delbrück Center for Molecular Medicine, Berlin, Germany, [3] Laboratory of Molecular Tumor Pathology, Institute of Pathology, Charité Universitätsmedizin Berlin, Berlin, Germany, [4] Laboratory of Bioorganic Synthesis, Department of Chemistry, Humboldt-Universität zu Berlin, Berlin, Germany

Systems biology studies the structure and dynamics of biological systems using mathematical approaches. Bottom-up approaches create models from prior knowledge but usually cannot cope with uncertainty, whereas top-down approaches infer models directly from data using statistical methods but mostly neglect valuable known information from former studies. Here, we want to present a workflow that includes prior knowledge while allowing for uncertainty in the modeling process. We build not one but all possible models that arise from the uncertainty using logical modeling and subsequently filter for those models in agreement with data in a top-down manner. This approach enables us to investigate new and more complex biological research questions, however, the encoding in such a framework is often not obvious and thus not easily accessible for researcher from life sciences. To mitigate this problem, we formulate a pipeline with specific templates to address some research questions common in signaling network analysis. To illustrate the potential of this approach, we applied the pipeline to growth factor signaling processes in two renal cancer cell lines. These two cell lines originate from similar tissue, but surprisingly showed a very different behavior toward the cancer drug Sorafenib. Thus our aim was to explore differences between these cell lines regarding three sources of uncertainty in one analysis: possible targets of Sorafenib, crosstalk between involved pathways, and the effect of a mutation in mammalian target of Rapamycin (mTOR) in one of the cell lines. We were able to show that the model pools from the cell lines are disjoint, thus the discrepancies in behavior originate from differences in the cellular wiring. Also the mutation in mTOR is not affecting its activity in the pathway. The results on Sorafenib, while not fully clarifying the mechanisms involved, illustrate the potential of this analysis for generating new hypotheses.

Keywords: systems biology, logical modeling, model checking, constraint based modeling, signaling pathways

OPEN ACCESS

Edited by:
Matteo Barberis,
University of Amsterdam, Netherlands

Reviewed by:
Julio Saez-Rodriguez,
European Bioinformatics Institute (EMBL-EBI), United Kingdom
Brian Paul Ingalls,
University of Waterloo, Canada

***Correspondence:**
Kirsten Thobe
kirsten.thobe@fu-berlin.de

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: *09 February 2018*
Accepted: *04 September 2018*
Published: *09 October 2018*

Citation:
Thobe K, Kuznia C, Sers C and Siebert H (2018) Evaluating Uncertainty in Signaling Networks Using Logical Modeling. Front. Physiol. 9:1335. doi: 10.3389/fphys.2018.01335

1. INTRODUCTION

Logical modeling has been shown to be a powerful tool for representing and analyzing biological systems (Saez-Rodriguez et al., 2007; Wang et al., 2012; Grieco et al., 2013). The main advantage in comparison to the standard modeling formalism in systems biology, Ordinary Differential Equations (ODE) modeling, is the low number of parameters, therefore logical models are mainly used to build large models that would be too complex for ODEs (Abou-Jaoudé et al., 2016). These models are usually built in a bottom-up manner, which means all available information about the

system is gathered and validated on new data (**Figure 1B**). A main issue when building these models is that uncertainty cannot be included into the model, e.g., if an influence between two components is controversial in the literature. Since only one model is created, the modeler needs to make an assumption neglecting the uncertain information. A second popular strategy for modeling is to use a top-down approach where the model is inferred directly from data, but here prior knowledge about the system is neglected (De Smet and Marchal, 2010).

As a consequence, alternative approaches have become more popular, where uncertainty is included into the modeling process by either building more than one model or by adapting the model through training. For exploring a wide range of models that is able to show a certain dynamical behavior, called family or pool of models, there are different methods available. The group of Saez-Rodriguez et al. developed a software *CellNOptR* to train a candidate model to data, accounting for topological uncertainties (Terfve et al., 2012). The output is a family of models selected for an optimality criterion, but cannot guarantee completeness due to stochastic search. The software *caspo* by the group of Siegel et al. uses Answer Set Programming to infer a family of logical models from experimental data based on optimization, where a tolerance accounts for experimental noise. The resulting family of models then represents all optimal models that reproduce the data and the software provides several analysis tools to explore properties of the models, such as classification for input/output behavior or experimental design (Videla et al., 2017). A similar method using time series data for inferring a model pool showed to be more precise than *caspo* (Ostrowski et al., 2016). A different approach was developed by our group, where uncertainty in parameters of the model, such as an uncertain sign of an edge, is encoded into the model definition (Klarner et al., 2012) and all possible models that arise from this uncertainty are enumerated. Subsequently the models are tested for satisfiability for data without an optimality criterion (**Figure 1A**), which was implemented using efficient formal verification techniques in *Tremppi* (Streck et al., 2015) and in *TomClass* (Klarner, 2014). Even though we employ the software from our group for the analysis in this paper, one could apply different software along the pipeline for building the model pool or analyzing it.

While computing model pools and testing them for data sets is computationally challenging, the analysis of potentially hundreds or thousands of models is not straight-forward in terms of the biological interpretation. Thus we propose a hypothesis-driven approach for specific biological questions, where the use of model pools allows us to test multiple hypotheses at the same time and analyze their interdependencies. Mathematical models are artificial constructs used to help understanding biological processes. In order to receive meaningful results from a modeling study, the biology needs to be transferred into mathematics and the results need to be interpreted from a biological perspective. In this paper, we address this task of incorporating biological information into the formalism by expanding the workflow in **Figure 1** to a four-step pipeline. At first, the process of bottom-up model building formalizes the biological phenomena into a prior knowledge network, which we call *system initialization*. Here, the regulatory graph and the logical equations are derived from literature information. Then, the *objective formalization* includes the aim of the study into the model setup, e.g., by adding extra components or edges. After generating the model pool, the top-down filtering process uses biological data that is not restricted to be of a specific type such as steady-state or input-output behavior. However, it requires a *data formalization* step. Finally, the *pool analysis* examines the specific pool for new biological insight.

In previous work, we presented parts of this pipeline, i.e. the objective of investigating crosstalk between two signaling pathways in Thobe et al. (2014), as well as challenges for data discretization and analysis in Streck et al. (2015) in context of a specific software. Here, we generalize and expand this pipeline by two additional objectives and analysis methods. Especially in the context of signaling processes in cancer cells, the identification of driver mutations is of great interest (Bozic et al., 2010), thus one aim of our framework is to identify driver mutations by a change in the logical function. The second aim presented is testing the effect of drugs by introducing them as new inputs to the system. Analyzing pools containing possibly hundreds or thousands of models is challenging. Here, we show a classification analysis to structure the resulting models toward interesting features, as well as extracting minimal mechanisms for a more detailed view on the models.

We apply this pipeline to model two central signaling processes involved in cancer, the mitogen-activated protein kinase (MAPK) cascade and the mTOR pathway (Shaw and Cantley, 2006; Saini et al., 2013), in two renal cancer cell (RCC) lines. Both cell lines were treated with the Raf-inhibitor Sorafenib

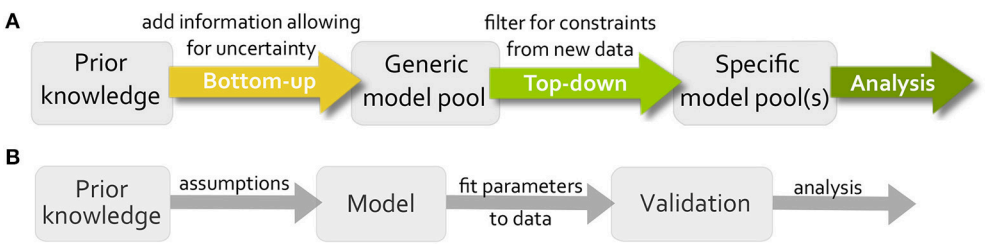


FIGURE 1 | Workflow for our modeling approach in comparison to traditional modeling process. **(A)** First a generic model pool is created from all available information including uncertainty. Then, the pool is filtered for data to find specific subpools, which can be analyzed for new information. **(B)** The traditional workflow creates one model based on assumption, estimates parameter values by fitting the model to data and validates the model on additional data.

yet displayed a differential response in terms of apoptosis induction (Kuznia, 2015). We hypothesized that the difference between these cell lines might be caused by distinct wiring of MAPK and mTOR signaling, which were shown to be connected via crosstalk (Mendoza et al., 2011; Aksamitiene et al., 2012). A rich dataset of time series measurement of key components in both pathways was generated using a high-throughput method, which was the foundation for the complex analysis presented in this study.

This paper is organized as follows. The Methods section first gives a brief introduction on the logical modeling framework and a detailed description on the pipeline we developed. In the Results section, the application on a signaling network is demonstrated, where first the model building process with the corresponding biological background is given, the data processing procedure is described and the results of the analysis are presented. Additionally, the biological interpretation is discussed and future experiments are suggested to wrap up the application section. Finally, the Discussion section exploits advantages and shortcomings of the method showing potential future extensions.

2. METHODS

2.1. Theoretical Background

The formalization of logical modeling for biological systems was introduced by Kauffman (1969) and further refined by Thomas (1991), which is the base for our work. However, we expanded this formalism to incorporate uncertain information leading to model pools (Klarner, 2014; Thobe et al., 2014, 2017; Streck, 2015; Streck et al., 2015).

2.1.1. Logical Modeling

The topology of a biological system is defined as a directed graph $\mathcal{R} = (V, E, l)$, called *interaction graph* (IG), where the nodes $V = \{1, ..., n\}$ represent the *components* of the system that are connected by edges $e \in E \subseteq V \times V$ called *interactions*, which represent a regulation of one component by another. The components adapt discrete values, called *activity levels*, and we consider *Boolean networks* (BN) with two levels assigned to each component $\mathbb{B} = \{0, 1\}$, where 0 means inactive and 1 stands for active. By assigning activity levels to every component of the network, the *state* of the system s is defined by $s : V \to \{0, 1\}, \forall v \in V : s(v) \in \mathbb{B}$. Here, the notation of a state is specified as a sequence in the order of V. In our approach, we add information about the nature of a regulation to each interaction using edge labels $l : E \to \{+, -, \neg+, \neg-\}$ (adapted from Klarner, 2014). In application, the labels $\{+, -\}$ are assigned to edges that represent well-known information, e.g., textbook knowledge, and are therefore required to be present in every model, which we call *essential*. In contrast, the labels $\{\neg+, \neg-\}$ are assigned to interactions that carry uncertainty, i.e., we not sure whether this interaction is present or not, which we call *optional*. However, we assume that the sign of an edge is known and exclude edges with unknown or ambivalent sign due to complexity.

Having defined the wiring of the network, the regulation of a component by its predecessors is defined by a logical function. The conditions describing when a component becomes active can be expressed using the logical operators $\vee$ (OR), $\wedge$ (AND), and $\neg$ (NOT) in a formula f_i for every component $v \in V$ consistent with the edge labels. This means that variables j are literals in f for component i, if $j \to i$ is a possible edge. Then a positive edge label has to cause an increasing value in the target component at some point, whereas a negative edge label has to cause a decrease. For optional edges, the increase or decrease can occur or that value is constant. However, in case the regulation of a component is uncertain of a component has optional incoming edges more than one model can be build from the available information. Then the set or all logical equations that are consistent with the edge labels are created and form the so-called *model pool*. An example is given in **Figure 2**.

2.1.2. Dynamical Behavior and Model Checking

In order to compare biological measurements with the dynamic behavior of the models, we need to define the transition from one state to another to generate the systems behavior over discrete time steps. For this aim, different update strategies have been developed, where some make assumptions on the timing of events, e.g., in synchronous update all components change in one transition, and others restrict the ordering of events, e.g., stochastic updates randomly update a component. Here, we employ asynchronous update, which is the least restrictive strategy at the cost of being computationally expensive (Thomas, 1991). In this strategy only one component can change its value per transition step, which means for $f_v(s) = \neg s_v$ for a state $s = (s_1, \ldots, s_v, \ldots, s_n)$ denote with $\bar{s}^v = (s_1, \ldots, \neg s_v, \ldots, s_n)$ the state which differs from s in the value of the component v. If no component changes $f(s) = s$ a steady-state of the system is reached. This update schedule produces every possible trajectory emerging from a state, thus the dynamics are non-deterministic which can be visualized in the so-called *state transition graph* (STG). Here, the states are the nodes of the graph and the transitions are edges.

After building the model pool from the available information, we want to filter those model that are in agreement with observed experimental data. Depending on the utilized software, either the data can be implemented as continuous values (e.g., Terfve et al., 2012) or needs to be discretized. Here, we want to describe two different kinds of biological data: time-series measurements and steady state observations. For this aim, we use *temporal logics* that are able to describe an ordering or a sequence of events in time, where *computation tree logic* (CTL) can cope with non-deterministic sequences (Clarke et al., 1986) and are therefore suited to explore the STG. For time-series measurements we encode a series of states that should exist at some point in the future and for steady states we encode the state of a component(s) that should hold for every state in the future. These formulas are then tested on the STG of the models using *model checking*. This process can be computationally expensive, since the state space exponentially increases with the number of components, also the number of models can quickly add up to thousands of models. For this reason, an efficient model checking software should be employed, e.g., Tremppi (Streck et al., 2015) and TomClass (Klarner, 2014).

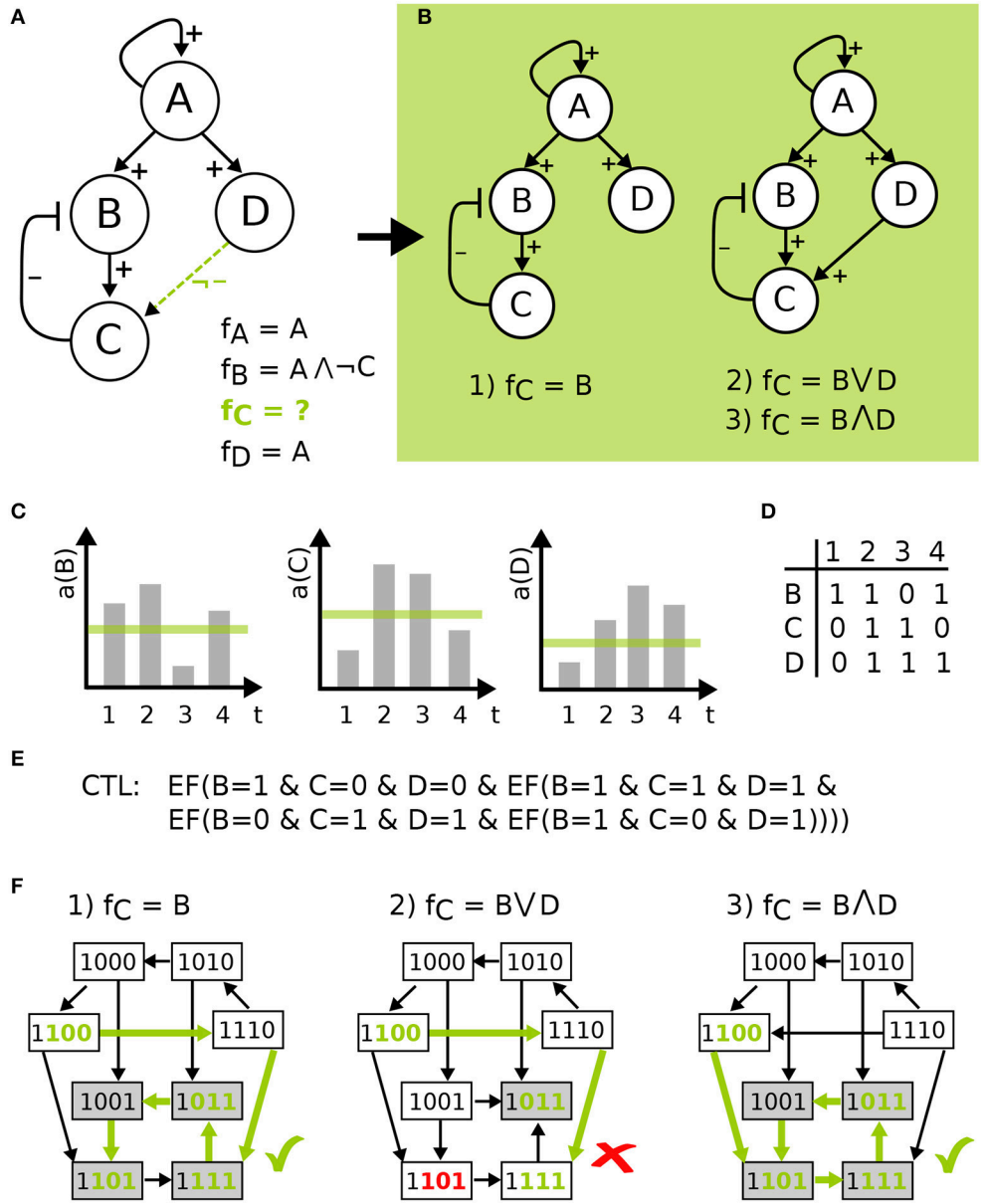


FIGURE 2 | Model definition to model checking visualized on a toy example. **(A)** IG with four components, edge labels and the corresponding functions resulting in model pool of size three in **(B)**. **(C)** shows time-series measurements for some activity *a* of B, C, and D, which is discretized by a threshold shown in green. The table in **(D)** gives the discretized data for the four time points, which are encoded as CTL formulas in **(E)**, where `EF(X)` is a CTL operator *exists finally*. This states that on some path from an initial state the `X` holds true at some point. STGs in **(F)** of the three models in the pool show the process of model checking for the CTL formula indicated by green states and edges, where the second model is not in agreement with the data.

2.1.2.1. Toy example

In **Figure 2**, model definition to model checking is visualized for a toy example. Here, an IG with four components is given, where the regulation of components A, B, and D is known, indicated by the edge labels and the corresponding functions. Component C has an uncertain regulation by component D, therefore the edge is labeled as not inhibiting and the function for C is undefined (**Figure 2A**). The resulting model pool then contains three different models that arise from the edge label. The process of temporal encoding of data is shown for time-series measurements, which is discretized by a threshold and encoded as CTL formulas, where the CTL operator *exists finally* is used. This operator states that the measurements must lie on one path in the ordering of the measurements in the STG, where there is no restriction made on how many states are visited in between the measurements. The CTL language offers more operators that could be employed depending on the type of data (Klarner et al., 2012), e.g., reflecting that the measurement frequency was so

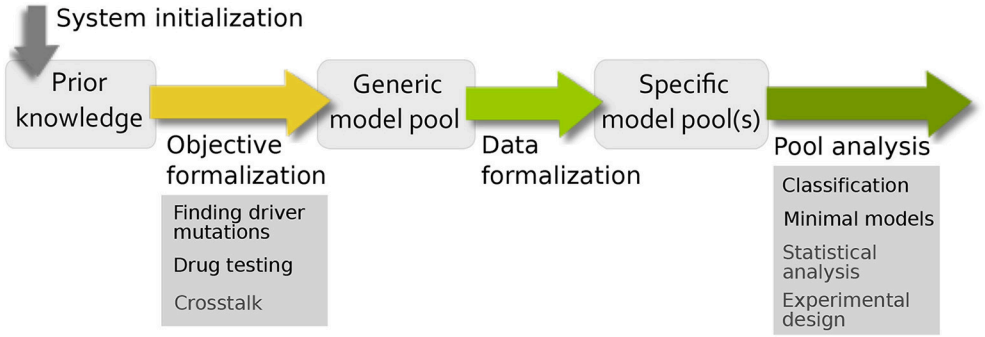


FIGURE 3 | Pipeline for evaluating uncertainty in biological systems. For building the prior knowledge network and defining the uncertainties of the system, the system initialization and objective formalization is necessary. The filtering process from the generic model pool to the specific model pool requires data formalization and the interpretation of the final pool is done by pool analysis.

high that one can assume that all qualitative activity changes of each component have been captured. However, in this paper we used the most conservative form. Finally, the process of model checking is visualized in **Figure 2F** showing that one model is not in agreement with the data.

2.2. Pipeline for Modeling Uncertain Systems

Based on the very general workflow in **Figure 1**, we want to formalize in more depth how specific biological questions can be addressed using model pools (**Figure 3**). In comparison to the traditional workflow for building and analyzing one model, there are similarities and differences. The main difference is that the standard bottom-up approach creates one model to test a single hypothesis, which is validated using (formalized) data and subsequently analyzed. Using model pools, the system initialization and data formalization remains the same, but we can test multiple hypothesis at the same time. This leads to a higher complexity in both the formulation of the aim of the study and the analysis of the model pool for biological information. To this end, the workflow has been adopted and specified to address common analysis themes for signaling networks. The resulting pipeline, shown in **Figure 3**, contains four steps: system initialization, objective formalization, data formalization, and pool analysis.

In the following, we provide a detailed formal description for the objective formalization and pool analysis. We assume that for the system initialization, available information is gathered and classified according to the theory presented in Section 2.1 for components, edges, and edge labels. Moreover, insight on regulations of components can be included by defining logical functions. This first step results in the *prior knowledge network* (PKN), which forms the starting point for our analysis (Saez-Rodriguez et al., 2011).

2.2.1. Objective Formalization

In the second step of our pipeline, we want to include the objective of the analysis into the PKN. In a simple setup, this could mean adding optional edges as hypotheses, but there are also more complex aims that require changes in the PKN. The first objective we identified, was to examine crosstalk between two pathways while preserving the dynamical properties of each pathway, presented in Thobe et al. (2014). Here, we want to present two different objectives: Finding driver mutations and drug testing.

2.2.1.1. Finding driver mutations

Cancer cells often accumulate mutations that are distinguished as either driver or passenger mutations. A lot of effort has been made to identify the driver mutations, since they are assumed to be a major cause for cancerous behavior (Greenman et al., 2007; Bozic et al., 2010). This abnormal behavior is due to the fact that the mutations affect the protein they are encoding in quality, changed sequence of the protein, or quantity, such as overexpression or knock-out of a gene. These effects cause changes in the regulatory network leading to an insensitivity of the component from its regulators, e.g., constantly active receptors. We aim to identify these changes in the regulation of a component in our approach, which we were able to confirm in previous work (Streck et al., 2016).

We account for mutations in components of a model with uncertain effect by setting the respective incoming and outgoing edges to optional even if these connections are textbook knowledge. In case more detailed information on the effect of the mutation is available, only a subset of edges can be set to optional. Formally, the network $\mathcal{R} = (V, E, l)$ is defined as in Section 2.1, where the set of mutated components is given by $V^m \subseteq V$ with edges $E^m = \{(u, v) \in E \mid v \in V^m \vee u \in V^m\}$. The labeling l^m of edges in E^m is set to:

$$l^m(u,v) = \begin{cases} \neg + & \text{if } (u,v) \in E^m \text{ and } l(u,v) = - \\ \neg - & \text{if } (u,v) \in E^m \text{ and } l(u,v) = + \end{cases}.$$

Thus, the affected edges are allowed to either stay the same or lose their function in the resulting model pool. If in the specific pool an incoming edge is not observable in any model, the mutated component becomes independent from its inputs and the function of the component can either be set to 0 or 1

indicating a loss-of-function or constitutively active mutation, respectively. A lost outgoing edge of the mutated component can indicate that the mutation affected the protein structure which can result in a dysfunctional protein. However, we do not account for gain-of-function mutations in this set-up, since this would require to add new edges to the model or change the sign of an edge. This would strongly increase the complexity of the study and should be addressed only based on suggestive data in a case by case way, which is no fit for this general set-up.

2.2.1.2. Drug testing

This objective aims to test qualitative effects of drugs on pools of models, without knowledge of the "true" network. Especially in cancer research, combinatorial therapies have become increasingly popular to enhance efficiency and overcome resistances (Ho et al., 2012; Manchado et al., 2016). Since we cannot represent concentrations or generally quantitative effects, the questions we want to address can be formulated as: where do we have to interrupt the signaling process to achieve a certain outcome. A similar study was done by Klinger et al. where they predicted the model structure and treatment quantitatively, however, the predictions resulting from the study were qualitative nature (Klinger et al., 2013).

For this approach, we introduce drugs as new components to the PKN and connect them with an inhibitory edge to their target, since they are supposed to suppress the activity of their target. For the network $\mathcal{R}' = (V', E', l')$ with f' as given logical equations, an extended set of components V is given by $V' \cup V^D$ where $v^D \in V^D$ is a set of drug components. The interactions of the network are given by $E = E' \cup E^D$ where new edges E^D are added, which contain an edge for self-activation for each new component to create the drug as input and an inhibitory edge from v^D to its target u, since the drug suppresses the activity of its target. Similarly, the set of labels is composed of the labels of the original network and the additional labels for the drug components, where known interactions are labeled with an essential label and uncertain effects with an optional label.

We can also include available information about the drug's mode of action into the logical function of its target. Usually drugs are selected to have a dominant influence on their target, for example through binding or modification it fails to interact with its former regulators. In case the logical equation of a drug target is known, we can directly translate this dominant effect on the target u in a new logical equation:

$$f_u = f'_u \wedge \neg v_D.$$

However, if detailed information about the biochemical properties of the drug on the target and other regulators is missing, the logical equation of the target is not defined and all possible regulations are generated in the pool.

2.2.2. Pool Analysis

After building the generic model pool from the PKN, this pool of models gets reduced for those models that are valid for data. Depending on the software, the data needs to be processed to apply it to logical models usually by discretization (Dimitrova et al., 2010; Gallo et al., 2015). As a result, we receive one or more specific model pools that need to be analyzed. For this aim, different kinds of analysis tools can be employed depending on the aim and the size of the resulting model pool such as statistical analysis (Thobe et al., 2014; Streck et al., 2016) or optimization (Terfve et al., 2012; Videla et al., 2017). Here, we want to present an analysis approach that allows a closer look at classes of models as well as single models.

2.2.2.1. Classification

Depending on the study, this pipeline can lead to specific model pools that contain too many models to analyze them by hand. This analysis step aims to get an intuition for commonalities or differences of models within the model pool, with respect to properties of interest. For this goal, properties such as validity for data or presence of an optional edge can be annotated to each model by e.g., using a database. Then, we can group sets of models into classes and compare them according to these properties to find difference between sets of models, for example we could observe that two optional edges are present in the model pool but occur mutually exclusive.

Here, the model pool is stored in a database and SQL queries are used to classify the models. For the queries, properties or a list of properties can be used as a classifier and are defined in the parameter `Classes`. Also we can restrict the pool to a subpool using the parameter `Restriction`, where we can select models for their property, e.g., only including all models that carry an optional edge. Mathematically, the analysis finds subsets of models that have a non-empty intersection and computes the cardinalities of these sets (Klarner, 2014). For this aim, an SQL query is generated using statements of the form:

```
SELECT DISTINCT Classes FROM models WHERE Restriction,
```

where `SELECT DISTINCT` computes all combinations of labels, i.e., subsets, of the selected `Classes` in the database `models`, possibly restricted using `WHERE`. Additionally, `COUNT` is used to determine the cardinality of each subset, i.e., the number of models in a class later denoted as size of a class. It is possible that classes are empty if there exists no model in the pool with a particular label combination.

2.2.2.2. Minimal models

While the classification gives a broad overview on the model pool, we also want to look at single models in the pool. The selection of models can be motivated by the objective and the biological background, by the classification analysis or by general criteria such as minimality. The criterion minimal can be interpreted in different ways: structural minimality in terms of number of edges, functional minimality in terms of shortest logical equations, or models that require least number of transitions to fulfill data. Each minimality can be interesting to regard separately or in combination. Structural minimality is a common biological assumption, where the system is assumed to have evolved in an energetically optimal way and is therefore sparse. Along with the number of interaction partners, the complexity of the regulation formulated as logical function can be assumed to be rather simple. Previous studies often used fixed rules for creating these

functions, such as activation- inhibition function (Martin et al., 2007), or optimized for short logical function (Videla et al., 2017).

Technically, this analysis counteracts the problem of overfitting. In general, the more degrees of freedom are available to a system, the easier is can produce various dynamics, thus our method has a bias toward building dense models. It is therefore beneficial to identify minimal structures or functions.

2.2.2.3. Interpretation of analysis results

Finally, the results from the analysis need to be transferred and interpreted to gain biological insight, which is not straight forward. Since the models are qualitative, the level of abstraction is high and the fact that we are looking at pools of models increases the complexity. However, by specifying clear objectives and predefined analysis options, the pipeline guides the modeling process and can deliver valuable information for experimental design or further modeling steps (Streck et al., 2015; Thobe et al., 2017).

3. RESULTS

3.1. Application on Growth Factor Signaling in Renal Cancer Cells

After presenting a pipeline to build model pools for different objectives and analysis options, we wanted to apply this pipeline to model growth factor signaling in two renal cancer cell lines, MZ1851RC and MZ1257RC. Motivation for this study was an observation that cell line MZ1851RC showed apoptosis after being treated with the drug Sorafenib while MZ1257RC seemed to be resistant (Kuznia, 2015). Sorafenib was developed to inhibit pathways controlling proliferation and cell survival and was shown to have anti tumor activity in colon, breast, and non-small lung cancer (Wilhelm et al., 2004; Gadaleta-Caldarola et al., 2015). The multikinase inhibitor Sorafenib was designed to suppress activity of Raf kinases in the MAPK pathway (Liu et al., 2006), however, it also affects a wide variety of receptor tyrosine kinases (RTKs) (Wilhelm et al., 2004). Very recently, Sorafenib was shown to inhibit the IGFR *in vitro* (Yaktapour et al., 2013), which indicates that Raf comprises an uncertain drug target in the renal cancer cell lines tested with our approach.

A second uncertainty was introduced by a mutation in the component mTOR in cell line MZ1851RC, but the effect of this mutation is unknown (Kuznia, 2015). A third uncertainty was caused by crosstalk between the MAPK pathway and PI3K signaling (**Figure 4A**), which was shown to compensate drugging of one of the pathways (Mendoza et al., 2011; Aksamitiene et al., 2012). Thus, the overall aim of this study was to clarify if the deviating behaviors are caused by differences the cellular wiring, which effect the mutation has and which targets Sorafenib is affecting.

3.2. Objective Formalization

The objective of this study splits into three different aims: investigating crosstalk between MAPK and PI3K pathways, finding the target for Sorafenib, and clarifying the effect of the mutation in mTOR. The PKN was extracted from literature, where the MAPK model was based on work by Kholodenko (2000) and the PI3K model was adapted from Courtney et al. (2010), also it is an adaption from a previous study (Thobe et al., 2014). For investigating the wiring between MAPK and PI3K pathway, candidate crosstalks were added. In detail, strongly activated MAPK signaling was found to cross-activate PI3K signaling, i.e., Erk was observed to phosphorylate Tsc2 suppressing it and Erk was also shown to phosphorylate Raptor, where both crosstalks activate mTORC1 signaling similarly to Akt (Roux et al., 2004; Winter et al., 2011). For simplicity, we summarized this effect to one crosstalk. Moreover, a cross-activation of EGFR on PI3K through Ras was shown, which is downstream of EGFR and upstream of Raf (Wong et al., 2010). A study of Will et al. found that PI3K inhibition, but not Akt inhibition, causes rapid decrease in wild type Ras activity and in Raf/Mek/Erk signaling concluding that PI3K cross-activates the MAPK cascade (Will et al., 2014). For the PKN, the crosstalk edges were labeled as optional edges and the edges within a pathway were assumed to be essential, shown in **Figure 4B**.

In order to test the effect of Sorafenib, it was added as additional input to the system as well as optional edges to possible targets: Raf, EGFR, and IGFR. Note that EGFR as Sorafenib target is a hypothesis and not based on experimental data. Moreover, one cell line, MZ1851RC carries a mutation in mTOR with unknown effect for mTORC1, thus the outgoing edge to IGFR was set to optional. A full list of optional edges in given in **Figure 4D**, also for some components the logical function can be set, since they only have one regulator (**Figure 4C**). All other components have undefined logical functions, which gives rise to the generic model pool.

Moreover, components that were neither measured nor perturbed were excluded from the model to reduce the complexity. For example, Mek and Tsc were not considered in the model, since both were lined up in a cascade as components with single input and output, thus deleting them does not pose problems for the model dynamics.

3.3. Data Formalization

3.3.1. Experiments Show Differential Behavior of Cell Lines

For our investigation, we used two different data sets: Western blot measurements of mTORC1 activity over time and a high throughput assay both published in Kuznia (2015). In the western blot measurements, the activity of mTORC1 was measured by its targets p70S6K (S6K) and S6RP in MZ1257RC and MZ1851RC cells. Here, the cells were either treated with DMSO or Sorafenib and the phosphorylation of the mTORC1 targets was measured over time. Regarding the measurements until 12 h, MZ1257RC cells showed a significant decrease in phosphorylation levels for S6K and S6RP. However, MZ1851RC cells only showed a reduction in S6RP phosphorylation for later time points, but the phosphorylation of S6K remained high. The 24 h time point is not considered, since we are only interested in signaling effects and this measurement is likely to be influenced by transcriptional effects. S6K was used as the read-out for the mTORC1 activity in the formal encoding of the Western blot data as CTL formulas WB.DMSO, WB1257Sora and WB1851Sora in the **Table 1**. Here, both cell lines show active mTORC1 for

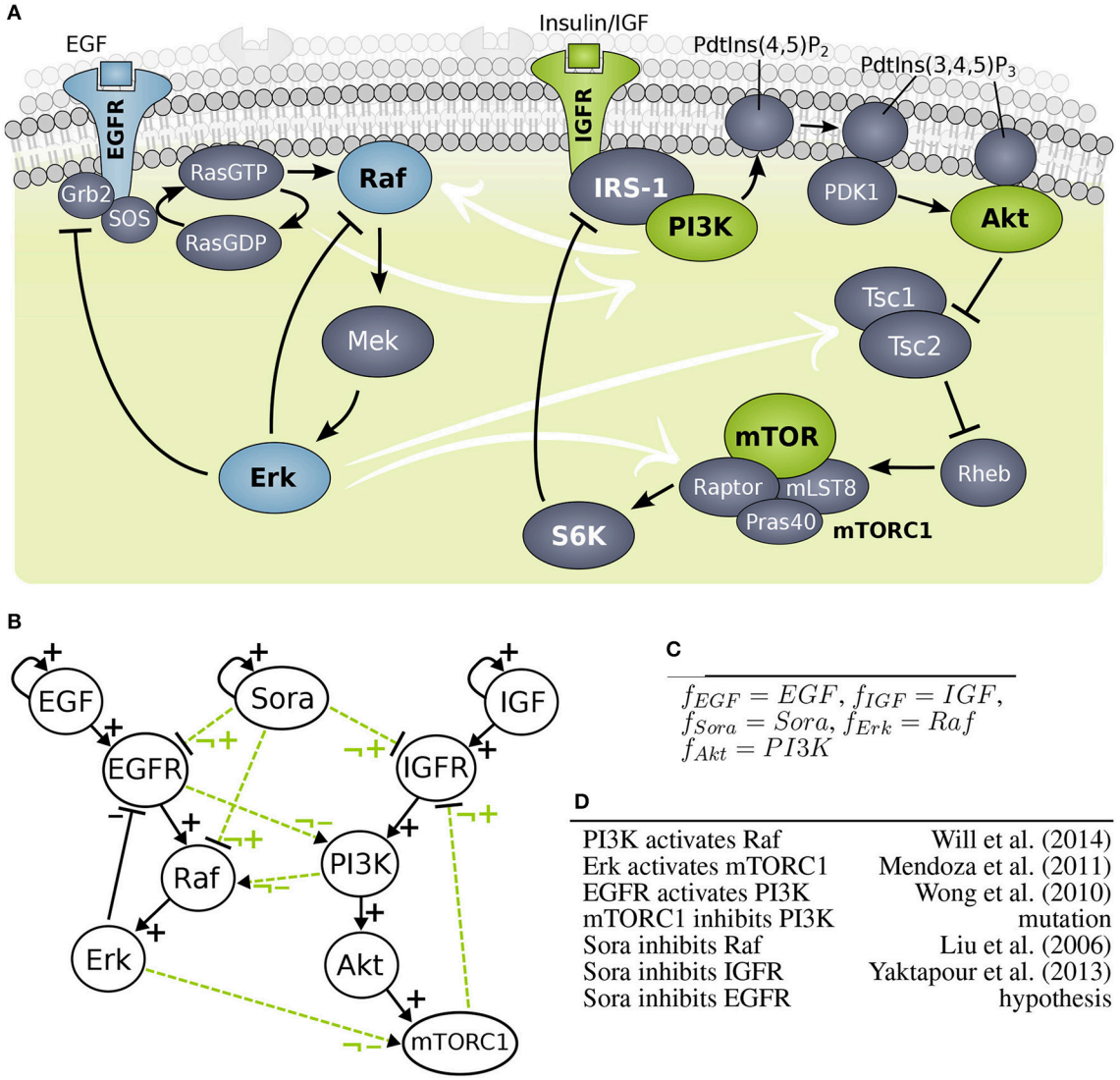


FIGURE 4 | Model building of growth factor signaling processes **(A)** Scheme of MAPK cascade and PI3K signaling. **(B)** Interaction graph of the MAPK (left hand side) and PI3K (right hand side) model marked with solid lines and optional influence of Sorafenib and crosstalk marked with dashed lines. **(C)** Predefined logical rules for regulations of components without optional incoming edges. **(D)** List of optional edges added to the network with references.

DMSO treatment throughout the measurements, thus a steady state was assumed and encoded in the CTL formula accordingly. For Sorafenib treatment, cell line MZ1257RC shows a steady state with decreased S6K phosphorylation, therefore mTORC1 was set to 0. In contrast, cell line MZ1851RC has stable S6K phosphorylation, thus mTORC1 was set to 1.

After observing differences in the activity of mTORC1 in the Western blots toward Sorafenib treatment, we wanted to investigate where the differences in the upstream regulation of mTORC1 originate from. For this aim, a high throughput approach using the Bio-Plex® system was applied (Kuznia, 2015). Here, the cells were unstimulated and not starved but treated with Sorafenib or DMSO and measured at different time points over a total period of 36 h in two experiments. In detail, the activity of the PI3K/mTORC1 signaling pathway was measured by the phosphorylation of Akt, and p70S6K as well as the MAPK activity was determined through the phosphorylation of Erk. Moreover, the receptors EGFR, and IGFR were included into the experiment, since we were interested whether the receptors are targeted by Sorafenib and to account for the feedback processes. For the complete dataset see Kuznia (2015), processed data is listed in the **Supplementary Table 1**.

3.3.2. Discretization of Time Series Data

In order to fit the models in to pool to the time series measurements, the data needs to be discretized. The choice of discretization method is influenced by the kind of data acquired and the experimental method used, for example with large data

TABLE 1 | Filtering model pool using model checking.

CTL formula	
WB.DMSO:	`EF(AG(mTORC1=1)) IS:Sora=0`
WB1257Sora:	`EF(AG(mTORC1=0)) IS:Sora=1`
WB1851Sora:	`EF(AG(mTORC1=1)) IS:Sora=1`
Bp1851Sora:	`EF(mTor=1&Akt=0&EGFR=0&Erk=0&IGFR=1&EF(mTor=1&Akt=1& EGFR=1&Erk=1&IGFR=1&EF(mTor=0&Akt=0&EGFR=0&Erk=0& IGFR=1))) IS:Sora=1`
Bp1851DMSO:	`EF(mTor=1&Akt=1&EGFR=1&Erk=1&IGFR=1&EF(mTor=1&Akt=1& EGFR=1&Erk=1&IGFR=0&EF(mTor=1&Akt=0&EGFR=1&Erk=1& IGFR=0&EF(mTor=0&Akt=0&EGFR=0&Erk=0&IGFR=0)))) IS:Sora=0`
Bp1851Sora2:	`EF(mTor=1&Akt=1&EGFR=1&Erk=1&IGFR=1&EF(mTor=0&Akt=1& EGFR=1&Erk=1&IGFR=1&EF(mTor=1&Akt=1&EGFR=1&Erk=1& IGFR=0&EF(mTor=1&Akt=0&EGFR=0&Erk=0&IGFR=0&EF(mTor=0& Akt=0&EGFR=1&Erk=1&IGFR=0&EF(mTor=1&Akt=0&EGFR=1&Erk=1& IGFR=1)))))) IS:Sora=1`
Bp1851DMSO2:	`EF(mTor=1&Akt=1&EGFR=1&Erk=1&IGFR=0&EF(mTor=0&Akt=1& EGFR=0&Erk=1&IGFR=0&EF(mTor=0&Akt=0&EGFR=0&Erk=1& IGFR=0&EF(mTor=1&Akt=1&EGFR=1&Erk=1&IGFR=0&EF(mTor=0& Akt=0&EGFR=0&Erk=0&IGFR=0&EF(mTor=0&Akt=0&EGFR=1&Erk=1& IGFR=0&EF(mTor=1&Akt=1&EGFR=1&Erk=1&IGFR=0)))))))) IS:Sora=0`
Bp1257Sora:	`EF(mTor=1&Akt=1&EGFR=1&Erk=1&EF(mTor=0&Akt=0&EGFR=0& Erk=0&EF(mTor=0&Akt=0&EGFR=1&Erk=0&EF(mTor=1&Akt=1& EGFR=1&Erk=1&EF(mTor=0&Akt=0&EGFR=1&Erk=0&EF(mTor=1& Akt=0&EGFR=1&Erk=1)))))) IS:Sora=1`
Bp1257DMSO:	`EF(Delta=0&mTor=1&Akt=1&EGFR=1&Erk=1) IS:Sora=0`
Bp1257Sora2:	`EF(mTor=0&Akt=0&EGFR=0&Erk=0&EF(mTor=0&Akt=1&EGFR=0& Erk=0&EF(mTor=1&Akt=1&EGFR=1&Erk=1&EF(mTor=1&Akt=1& EGFR=1&Erk=0&EF(mTor=1&Akt=0&EGFR=1&Erk=1&EF(mTor=1& Akt=1&EGFR=1&Erk=1)))))) IS:Sora=1`
Bp1257DMSO2:	`EF(mTor=1&Akt=0&EGFR=1&Erk=1&EF(mTor=1&Akt=0&EGFR=1& Erk=0&EF(mTor=1&Akt=1&EGFR=1&Erk=1&EF(mTor=0&Akt=0& EGFR=0&Erk=0&EF(mTor=1&Akt=1&EGFR=1&Erk=1))))) IS:Sora=0`

CTL formulas derived from Western blot and Bio-Plex® experiments. For denoting the CTL formulas, the following semantics are used: `EF(X)` *: is a CTL operator* exists finally. *This states that on some path from an initial state the* `X` *holds true at some point.* `AG(X)` *: is a CTL operator* all globally. *This states that X has to hold for all future states, i.e., X is in a steady state.* `v=b` *: where* $v \in V, b \in \mathbb{B}$ *states that value of a component* `v` *is set to* `b`. `IS` *: declares the initial state and is a list of boolean constraints on the values of the components. A state is considered initial, if all the constraints are satisfied.*

sets statistical methods provide good results, but with small data sets the choice is more difficult (Dimitrova et al., 2010). Here, we opted to show a simple approach by using the arithmetic mean as threshold. More specifically, the data was discretized by defining for each experiment e and component v, a threshold

$$\theta_{ev} = \frac{\sum_t \sum_x m_{evtx}}{|t||x|}$$

where, m_{evtx} is the measured activity of component v in the experiment e with treatment x. The total number of time points is $|t|$ and the total number of treatments is $|x|$.

Since the cells were cultivated and treated in parallel, the phosphorylated levels for both treatments were expected to be comparable. Thus, the threshold for e.g., Erk is the same mean value under both Sorafenib and DMSO treatment within each cell line for each experiment. Moreover, the standard deviation for each component was calculated in order to avoid the problem of discretizing a component that does not change over time. By looking at small standard deviations relative to the mean, IGFR measurements for MZ1257RC in both experiments were identified as problematic (see **Supplementary Table 1**). Comparing the IGFR levels between the cell lines, we decided to exclude this data.

Since we are interested in the signaling processes, only measurements until 8 h were included. The resulting CTL formulas are listed in **Table 1**, where all Bioplex measurements were encoded as transient states, due to the fact that they changed throughout the 8 h of measurement. An exception is the data set Bp1257DMSO, which was encoded as steady state (see **Supplementary Table 1** MZ1257RC-DMSO Exp1). Note that the discretization of data is not always straight-forward, thus we excluded data which was problematic mathematically (such as IGFR) or had poor quality in the measurements.

3.3.2.1. Robustness of results

As a basic test of robustness with respect to the discretization method being used, we additionally performed a discretization by median instead of mean value. This change in the discretization

TABLE 2 | Number of models consistent with CTL formulas.

CTL formula	Pool size
(A)	
WB.DMSO	15,026
WB1851Sora	5,902
Bp1851Sora	10,080
Bp1851DMSO	12,474
Bp1851Sora2	5,632
Bp1851DMSO2	7,216
Rp.1851	293
(B)	
WB.DMSO	15,026
WB1257Sora	15,026
Bp1257Sora	9,984
Bp1257DMSO	12,096
Bp1257Sora2	12,393
Bp1257DMSO2	10,032
Rp.1257	1017

Filtering for CTL formulas gives pool sizes as the number of models in agreement. Rp.1851 and Rp.1257 are the cell line specific pools as the intersection of the data sets shown in (A,B), respectively.

threshold had a negligible effect on both cell lines: 5.7% of the boolean values changed for MZ1257RC and 7.1% for MZ1851RC. Furthermore, we repeated the subsequent analysis using the median discretization, and observed only minor changes in the size of the model pool for cell line MZ1257RC and no change in the resulting biological interpretation of that pool.

3.4. Pool Analysis

After deriving the PKN from the literature and including the objectives of the study, the generic model pool was created. As a result from combining of all optional edges and logical equations the pool contains 19,404 models. In order to find biologically relevant models, the third step of the pipeline generated the specific pool(s) by filtering the generic pool for those models that are able to simulate experimentally observed behavior for the two RCC cell lines.

3.4.1. Cell Line Specific Model Pools

Each CTL formula has a non-zero pool size and is therefore feasible for our analysis (see **Table 2**). To determine the cell line specific models, we calculated the intersection of the different subpools for cell line MZ1257RC as Rp.1257 and for cell line MZ1851RC as Rp.1851:

- $Rp.1851 = WB.DMSO \cap WB1851Sora \cap Bp1851Sora \cap Bp1851Sora2 \cap Bp1851DMSO \cap Bp1851DMSO2$
- $Rp.1257 = WB.DMSO \cap WB1257Sora \cap Bp1257Sora \cap Bp1257Sora2 \cap Bp1257DMSO \cap Bp1257DMSO2$

Note that both pools are required to fulfill WB.DMSO, since this dataset was identical for both cell lines. Although the single CTL formulas resulted in relatively large pools, containing 5,000–15,000 models, the intersection for the cell line specific pools shows a strong reduction with 1017 models for Rp.1257 and 293 models for Rp.1851 (see **Table 2**). Thus, there exists a cell line specific pool for each cell line. One interesting question is whether these cell line specific pools share any models, which we addressed by calculating the intersection between Rp.1257 and Rp.1851. The result is an empty set, which means the model pools Rp.1257 and Rp.1851 are disjoint. In the next step, we wanted to further characterize and explore these cell line specific pools for information on crosstalk and Sorafenib targets.

3.4.2. Classification Shows Differences Between Cell Lines

Besides the sizes of the pools and the information about the intersection of subpools, we did not receive any information about the models within a pool yet. Since we were interested in the structure of the models, especially the wiring of Sorafenib and crosstalk edges, we selected the classification analysis from the pipeline. Here, we classified for the number of optional edges and the presence of an optional edges. As a result, all models within one class have the same interaction graph, thus only differ in their logical functions (see tables in online repository). Looking at these classes, we can state that for both specific pools there are no rejected edges, since each edge appears in at least one model. This also means that we cannot exclude any of the Sorafenib targets for both cell lines. We can visualize these results by showing the frequency of edges across a pool in **Figure 5**, where we define the frequency of an edge as the number of models containing this edge divided by the number of models in the pool. Here, the graphs (A) and (B) for the full pools of each cell line show differences in the frequency of the optional edges, especially in the Sorafenib targets and the feedback. While in MZ1257RC the frequency of Sora influences is high, in MZ1851RC they are low especially for IGFR and EGFR. Moreover, one of the objectives was to identify the effect of the mutation in mTOR on the feedback in cell line MZ1851RC, where in **Figure 5B** 100% of the models in the pool Rp.1857 contain this feedback, while in the other cell line this value only reaches 71%. Also we can observe that every optional edge is present, since no edge is missing.

We can also restrict the classification to a subset of models, e.g., shown in the **Figures 5C,D**. For cell line MZ1257RC, we selected all models that contain an activating edge from Sorafenib to Raf, where we can observe an enrichment in the crosstalk from PI3K to Raf compared to the full pool (**Figure 5C**). In contrast, if we filter for all models with an edge from Sora to EGFR in cell line MZ1851RC (**Figure 5D**), the connection between Sora and IGFR is lost, which means that there is no model containing both edges. Moreover, the frequency of the connection between EGFR and PI3K is reduced. However, since these effects are statistics across a pool of models, it is hard to draw any conclusions about single models.

3.4.3. Minimal Mechanisms for Sorafenib Targets

Although the classification analysis provides a good overview and intuition about the cell line specific pools, the result shows that a more detailed view can provide more information. Looking at

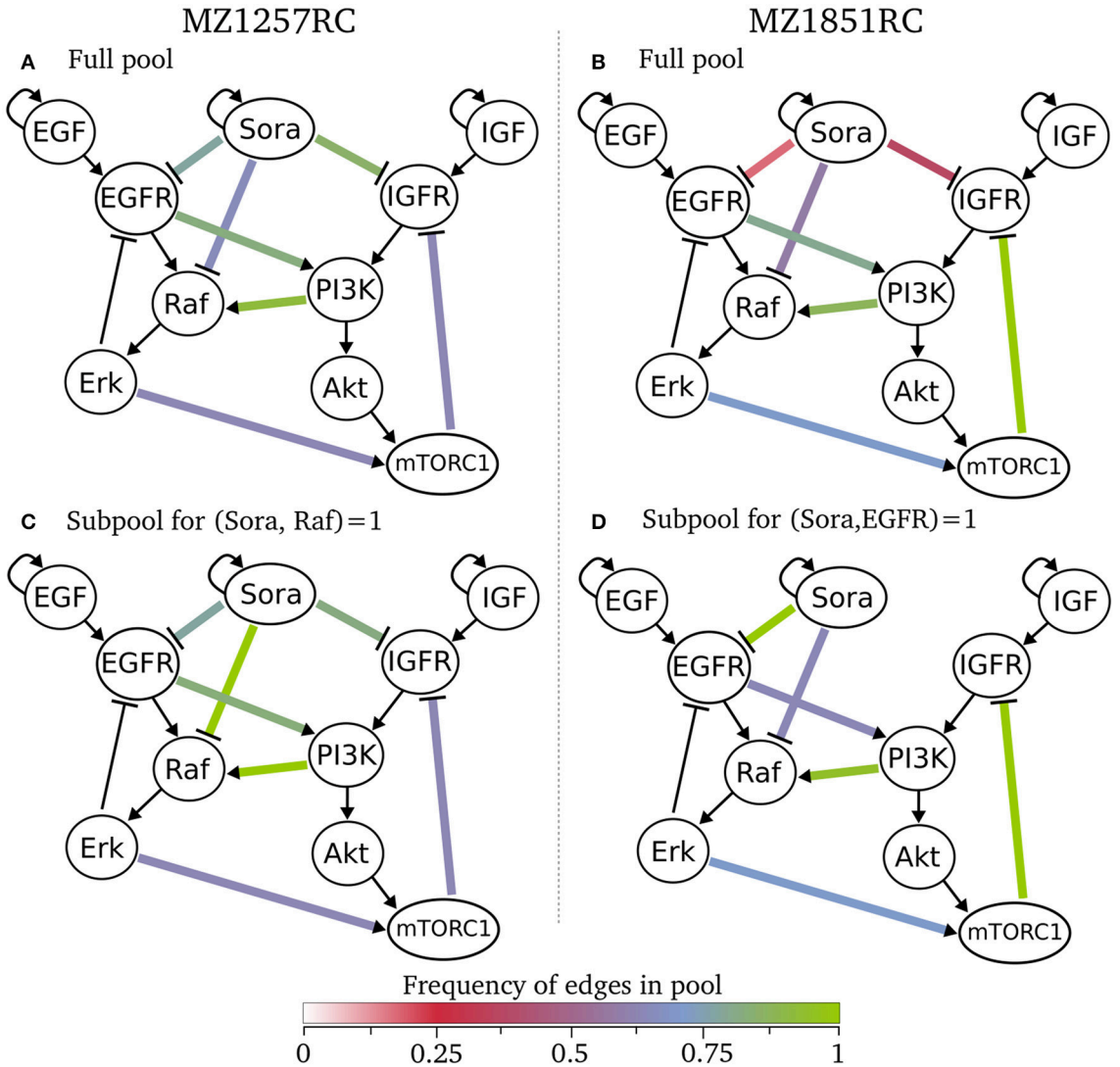


FIGURE 5 | Classification of specific pools for both cell lines, where the color of an optional edge gives the frequency in the respective subpool. In **(A)** all 1017 models for cell line MZ1257RC are visualized and in **(C)** the subpool of 193 models that contain the edge from Sora to Raf is shown. In **(B)** all 293 model of cell line MZ1851RC are presented and in **(D)** a subpool of 72 models with an edge from Sora to EGFR are shown.

the minimal structures or mechanisms of each pool, we wanted to extract more information on how the crosstalk might be linked to the Sorafenib mechanism. For this aim, we analyzed the cell line specific pools for two features: the number of Sorafenib targets and possible crosstalk mechanisms. Due to the large number of models in the pools, we separated the pool for three scenarios: no influence of Sorafenib, meaning that all three optional outgoing edges of Sora are not present, Sorafenib has one target only, Sorafenib has exactly two targets and Sorafenib has exactly three targets. In **Table 3**, the minimal models according to these scenarios for the pool Rp.1257 and in **Table 4** for the pool Rp.1851 are listed.

The specific pool for MZ1257RC shows that every combination of Sorafenib target from none to all is present in the pool, thus we cannot exclude any hypotheses in this cell line (see **Table 3**). However, we can see that every model contains at least one crosstalk edge and every edge appears in a minimal model. For one Sora target, there is always a basic crosstalk from the MAPK pathway to PI3K signaling either by EGFR on PI3K or by Erk on mTORC1. For Raf, additionally the crosstalk from PI3K on Raf becomes necessary, which we already identified in **Figure 5C**. For dual targets in **Table 3C**, IGFR/EGFR requires the cross-activation from EGFR on PI3K or the feedback, IGFR/Raf require (PI3K,Raf) in combination with any of the other crosstalk or the feedback and EGFR/Raf needs (PI3K,Raf) and one crosstalk. In case all three targets are affected by Sora, the (PI3K,Raf) edge and any of the crosstalks or the feedback are required.

The minimal structures in the second cell line MZ1851RC exclude two scenarios: IGFR/EGFR as dual targets (as shown in

TABLE 3 | Minimal mechanisms for Sorafenib targets and crosstalk in Rp.1257.

	Sorafenib targets	(EGFR, PI3K)	(Erk, mTORC1)	(mTORC1, IGFR)	(PI3K, Raf)
(A)	None	0	1	0	0
(B)	IGFR	1	0	0	0
		0	1	0	0
	EGFR	1	0	0	0
		0	1	0	0
	Raf	1	0	0	1
		0	1	0	1
(C)	IGFR/EGFR	1	0	0	0
		0	0	1	0
	IGFR/Raf	1	0	0	1
		0	0	1	1
		0	1	0	1
	EGFR/Raf	1	0	0	1
		0	1	0	1
(D)	All	1	0	0	1
		0	0	1	1
		0	1	0	1

Minimal models after classification of Rp.1257 for (A) no Sorafenib targets, (B) exactly one target, (C) exactly two targets, and (D) all three possible targets are affected.

TABLE 4 | Minimal mechanisms for Sorafenib targets and crosstalk in Rp.1851.

	Sorafenib targets	(EGFR, PI3K)	(Erk, mTORC1)	(mTORC1, IGFR)	(PI3K, Raf)
(A)	None	0	1	1	0
		1	0	1	0
(B)	IGFR	1	0	1	0
		0	1	1	0
	EGFR	1	0	1	0
		0	1	1	0
	Raf	1	0	1	1
		0	1	1	1
(C)	IGFR/EGFR				
	IGFR/Raf	0	1	1	1
		1	0	1	1
	EGFR/Raf	0	1	1	1
		1	0	1	1
(D)	All				

Minimal models after classification of Rp.1851 for (A) no Sorafenib targets, (B) exactly one target, (C) exactly two targets, and (D) all three possible targets are affected.

Figure 5D) and all three targets simultaneously. **Table 4** shows models in the pool that are not affected by Sorafenib. Compared to cell line MZ1257RC, there are similarities and difference in the model structures. Raf as a Sorafenib target again requires the edge from PI3K on Raf to be present and also the models always require a crosstalk from the MAPK pathway on PI3K signaling, but additionally the feedback is essential for every model.

3.4.4. Interpretation of Analysis Results

The minimal models give an overview about how the system could compensate the influence of the inhibitor to fit the data for different levels of influence. For this aspect the cell lines show similarities and differences in their model structures, where two trends can be extracted from the minimal mechanisms. First, all models require at least one crosstalk edge to be able to produce trajectories that match the data we applied. Interestingly, adding more Sorafenib targets most often does not enforce more or different crosstalk edges, with the exception of Raf. Within Rp.1851 the mechanisms for every Sorafenib target and all combinations show the identical minimal mechanism, plus the edge for Raf. A possible explanation for this observation could be the symmetrical structure of the model, in particular when the feedback is active as it is in every model of Rp.1851. Both pathways consist of a cascade of activating edges with a negative feedback on the Sorafenib target. It would be interesting to apply data which breaks with this symmetry, e.g., with a PI3K inhibitor to block the crosstalk.

The second clear trend we can identify from the results is that Raf as Sora target requires the cross-activation from PI3K on Raf. Since Raf is the designated Sorafenib target, this result is interesting. Looking at the PKN structure and the data, we can see that Erk becomes active under Sorafenib treatment and the only activator for Erk is Raf. In the MAPK pathway, Raf is activated by EGFR, which itself is inhibited by Erk. Thus, if Erk should become and stay active over longer time periods as shown in the data, Raf needs another activator to compensate the inhibition through Sorafenib. However, Sorafenib was described to have a paradoxical effect on the MAPK pathway. While, in cell lines carrying a BRAF mutation the signaling was efficiently blocked, cell lines with WT-BRAF showed an activation of Erk (Hatzivassiliou et al., 2010; Heidorn et al., 2010; Poulikakos et al., 2010). Thus, further investigations are necessary to exploit whether this observation is an artifact of the model or has biological relevance. In detail, paradoxical activation by Sorafenib and the role the crosstalk from PI3K to Raf would need to be examined, which would require a refined model where the edge from Sorafenib on Raf could also be activating and more data, e.g., an experiment with a PI3K inhibitor would be interesting in this context.

3.4.4.1. Overlap of Sorafenib targets

Another general question is, whether we assume Sorafenib to have the same targets in both cell lines. One could argue that the cell lines could differ in their internal wiring meaning the crosstalk, but the biochemical targets of Sorafenib should be independent of cell lines. Assuming that all three targets, IGFR, EGFR, and Raf, are expressed in both cell lines, the intersection of the results in **Tables 3**, **4** would further narrow down possible targets. In that case, we could exclude the case of Sorafenib affecting all targets simultaneously, since in cell line MZ1851RC there are no models that have IGFR, EGFR, and Raf as targets. Moreover, the combination IGFR/EGFR is not present in Rp.1851, thus either Sorafenib targets either one of the receptors by themselves or additionally Raf in these cell lines. Even though these results are not clear, they can support

and guide further studies, especially experiments where receptors are stimulated additionally to the drug treatment would be beneficial.

3.4.4.2. Models without Sorafenib targets

A surprising result of the analysis is the presence of models without a Sorafenib target. In cell line MZ1257RC, <1% of the models have no Sorafenib target, while the pool for MZ1851RC 16% of the models fall into this category. Since the data clearly shows an effect of the drug on components in this pathway, we expected all models to have at least one target of Sorafenib to be influenced. Thus, the data set from cell line MZ1851RC seems to be not restrictive enough for every model to require an interaction from Sorafenib. Since only a subset of components is measured, some models can match the data by specific initial states. Here, additional data would be beneficial to refine the results, especially measuring more components would reduce the degree of freedom for fitting the data.

4. DISCUSSION

In this paper, we present an alternative approach to standard modeling procedures. Instead of building and validating one model, we incorporate uncertain information or hypotheses to build a pool of models that is then filtered for data and analyzed using specific strategies. An advantage of this method is that we can test multiple hypotheses at the same time, but it comes at the cost of high complexity and challenging analysis. For this reason, we created a pipeline with specifically defined objectives and analysis templates that the modeler can select and combine. In addition to templates for objectives, data formalization and pool analysis presented in previous work (Thobe et al., 2014, 2017; Streck et al., 2015), we introduce two new objectives, namely finding driver mutations and drug testing, as well as two analysis options, namely classification and minimal models.

In the second part of the paper, the pipeline is applied to study the uncertain wiring and effect of a drug in cancer cells based on a rich data set. Two RCC cell lines, MZ1257RC and MZ1851RC, were observed to behave differently upon Sorafenib treatment, thus we tested possible drug targets in the MAPK and PI3K signaling and also investigated possible crosstalk between these pathways in a cell line specific manner, incorporating a mutation with uncertain effect as objectives. As a result, a substantial reduction from 19,404 for the initial pool to 1,017 for MZ1257RC and 293 for MZ1851RC was observable, and the empty intersection of both pools shows that the cell line specific models indeed have a different wiring. In order to cope with the complexity of having hundreds of models as outcome of the study, we developed different analysis tools. Here, we showed that classification of the pool can provide an overview on the models in the pool and give information on essential or neglected edges. In the case study, the classification showed that the feedback from mTORC1 on IGFR was active in both cell lines. We had set this edge to optional since the cell line MZ1851RC carries a mutation in mTOR and we hypothesized that this affects the feedback. As a result, all models in Rp.1851 show the feedback in their models and thus the mutation does not affect the function of mTORC1 toward IGFR. However, for cell line MZ1257RC, which does not carry a mutation in mTOR, only 71% of the models in the pool contain this edge. An explanation for this could be that we had to exclude the data for IGFR in the Bioplex experiment, since the variance of the data was too low to allow for a meaningful discretization, which could also be the reason for the larger model pool in comparison to MZ1851RC.

For the crosstalk and the Sorafenib mode of action the classification analysis showed no clear trend, since the results are complex and hard to interpret. For this reason, we listed the minimal models according to the number of Sorafenib targets and the required crosstalk to gain more detailed information on the simplest solution (**Tables 3**, **4**). Even though we cannot exclude any Sorafenib target and crosstalk in the analysis, we are able to identify patterns, where specific Sorafenib targets require different crosstalk edges to be present, e.g., Raf requires a crosstalk from PI3K on Raf. Another important observation from the classification was that there are models in the pools of both cell lines without any interaction between Sorafenib and its target. The conclusion from this result is that the data was not restrictive enough to exclude these models and further data is necessary resolve this issue. However, in case we would only fit one model to the data, we would have missed this lack of expressiveness.

The strength of underlying approach is based on its paradigm of considering possibly huge sets of models for testing and comparison. Consequently, it does not scale as well as single model approaches. The software utilized here is limited by its model checking tool NuSMV, where more than 50 components are not solvable within reasonable time. For the analysis presented in this paper, the program was run on a Ubuntu 17.10 workstation with a processor i7-7700, 3.6 GHz, and 32GB RAM. The script with 10 components took 143 minutes in TomClass, which included building the pool, model-checking, and classification of 19,404 models. Tools like caspo list running times of approximately 56 mins for models of size 45 generating a model pool with 384 models and thus can still handle medium sized models (Videla et al., 2017). The software Tremppi was shown to be able to handle a model pool of size 259,200 and perform model-checking of 40 data sets within 151–177 min depending on parameter settings on a similar workstation (Streck et al., 2016). In general, the kind of models that are feasible for this approach are a trade-off between number of components and number of uncertain edges, the latter of which affects the size of the model pool. This approach aims at exploring uncertainties in small to medium sized models, which is well-suited to represent interesting processes such as signaling pathways and regulatory modules.

While the generation and especially model-checking process is computationally expensive, the analysis and interpretation of these pools, which in our case are just large tables, is challenging from a biological perspective. Thus we propose to define clear objectives for designing the study as well as offer different analysis strategies to extract new information from the complex results. There are many possibilities for extensions especially one could think of further biologically

interesting objectives, but also including different kind of data and more analysis options such as algorithms to find special patterns. Moreover, we are not limited to the Boolean set-up, but are able to handle multivalued models (Streck et al., 2015). Finally, although the pipeline was developed for signaling networks, the approach can be applied to any related modeling problem.

DATA AND SOFTWARE AVAILABILITY

Python scripts, data sets and the classification of the model pools from the case study analysis are available on GitHub: https://github.com/kthobe/RCC_ModelPoolAnalysis.
The software TomClass is also available on GitHub: https://github.com/hklarner/TomClass.

AUTHOR CONTRIBUTIONS

KT designed study, analyzed and interpreted data, implemented computations, and drafted the manuscript. CK designed and performed experiments, processed and interpreted data, revised the manuscript. CS participated in discussion, interpreted data, and revised the manuscript. HS participated in conceptualization and discussion, interpreted data, reviewed and revised manuscript. All authors read and approved the final manuscript.

FUNDING

The work was partially funded by the German Federal Ministry of Education and Research (BMBF), grant no. 0316195.

ACKNOWLEDGMENTS

This work was based on the dissertation of Thobe (2017). The authors would like to thank Hannes Klarner and Adam Streck for technical support.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphys.2018.01335/full#supplementary-material

Supplementary Table 1 | Dataset from two Bioplex experiments. Each experiment was done for each cell line, where MZ1851RC is shown in the top table and MZ1257RC in the bottom. Both cell lines were treated with DMSO or Sorafenib at the indicated time before measurement and all measured at once, thus one experiment can be seen as one batch. Measurements marked in red were excluded from the analysis due to low variance.

REFERENCES

Abou-Jaoudé, W., Traynard, P., Monteiro, P. T., Saez-Rodriguez, J., Helikar, T., Thieffry, D., et al. (2016). Logical modeling and dynamical analysis of cellular networks. *Front. Genet.* 7:94. doi: 10.3389/fgene.2016.00094

Aksamitiene, E., Kiyatkin, A., and Kholodenko, B. N. (2012). Cross-talk between mitogenic Ras/MAPK and survival PI3K/Akt pathways: a fine balance. *Biochem. Soc. Trans.* 40, 139–146. doi: 10.1042/BST20110609

Bozic, I., Antal, T., Ohtsuki, H., Carter, H., Kim, D., Chen, S., et al. (2010). Accumulation of driver and passenger mutations during tumor progression. *Proc. Natl. Acad. Sci. U.S.A.* 107, 18545–18550. doi: 10.1073/pnas.1010978107

Clarke, E. M., Emerson, E. A., and Sistla, A. P. (1986). Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Trans. Programm. Lang. Syst.* 8, 244–263. doi: 10.1145/5397.5399

Courtney, K. D., Corcoran, R. B., and Engelman, J. A. (2010). The PI3K pathway as drug target in human cancer. *J. Clin. Oncol.* 28, 1075–1083. doi: 10.1200/JCO.2009.25.3641

De Smet, R., and Marchal, K. (2010). Advantages and limitations of current network inference methods. *Nat. Rev. Microbiol.* 8, 717–729. doi: 10.1038/nrmicro2419

Dimitrova, E. S., Licona, M. P. V., McGee, J., and Laubenbacher, R. (2010). Discretization of time series data. *J. Comput. Biol.* 17, 853–868. doi: 10.1089/cmb.2008.0023

Gadaleta-Caldarola, G., Infusino, S., Divella, R., Ferraro, E., Mazzocca, A., De Rose, F., et al. (2015). Sorafenib: 10 years after the first pivotal trial. *Fut. Oncol.* 11, 1863–1880. doi: 10.2217/fon.15.85

Gallo, C. A., Cecchini, R. L., Carballido, J. A., Micheletto, S., and Ponzoni, I. (2015). Discretization of gene expression data revised. *Brief. Bioinform.* 17, 758–770. doi: 10.1093/bib/bbv074

Greenman, C., Stephens, P., Smith, R., Dalgliesh, G. L., Hunter, C., Bignell, G., et al. (2007). Patterns of somatic mutation in human cancer genomes. *Nature* 446, 153–158. doi: 10.1038/nature05610

Grieco, L., Calzone, L., Bernard-Pierrot, I., Radvanyi, F., Kahn-Perlès, B., and Thieffry, D. (2013). Integrative modelling of the influence of MAPK network on cancer cell fate decision. *PLoS Comput. Biol.* 9:e1003286. doi: 10.1371/journal.pcbi.1003286

Hatzivassiliou, G., Song, K., Yen, I., Brandhuber, B. J., Anderson, D. J., Alvarado, R., et al. (2010). Raf inhibitors prime wild-type raf to activate the mapk pathway and enhance growth. *Nature* 464:431. doi: 10.1038/nature08833

Heidorn, S. J., Milagre, C., Whittaker, S., Nourry, A., Niculescu-Duvas, I., Dhomen, N., et al. (2010). Kinase-dead braf and oncogenic ras cooperate to drive tumor progression through craf. *Cell* 140, 209–221. doi: 10.1016/j.cell.2009.12.040

Ho, A. L., Musi, E., Ambrosini, G., Nair, J. S., Vasudeva, S. D., de Stanchina, E., et al. (2012). Impact of combined mTOR and MEK inhibition in uveal melanoma is driven by tumor genotype. *PLoS ONE* 7:e40439. doi: 10.1371/journal.pone.0040439

Kauffman, S. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.* 22, 437–467. doi: 10.1016/0022-5193(69)90015-0

Kholodenko, B. N. (2000). Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades. *Eur. J. Biochem.* 267, 1583–1588. doi: 10.1046/j.1432-1327.2000.01197.x

Klarner, H. (2014). *Contributions to the Analysis of Qualitative Models of Regulatory Networks*. Ph.D. thesis, Freie Universität Berlin.

Klarner, H., Siebert, H., and Bockmayr, A. (2012). Time series dependent analysis of unparametrized thomas networks. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 9, 1338–1351. doi: 10.1109/TCBB.2012.61

Klinger, B., Sieber, A., Fritsche-Guenther, R., Witzel, F., Berry, L., Schumacher, D., et al. (2013). Network quantification of EGFR signaling unveils potential for targeted combination therapy. *Mol. Syst. Biol.* 9:673. doi: 10.1038/msb.2013.29

Kuznia, C. (2015). *Molecular Mechanisms of Sorafenib-Induced Apoptosis in Cancer Cells*. Ph.D. thesis, Humboldt-Universität zu Berlin.

Liu, L., Cao, Y., Chen, C., Zhang, X., McNabola, A., Wilkie, D., et al. (2006). Sorafenib blocks the RAF/MEK/ERK pathway, inhibits tumor angiogenesis, and induces tumor cell apoptosis in hepatocellular carcinoma model PLC/PRF/5. *Cancer Res.* 66, 11851–11858. doi: 10.1158/0008-5472.CAN-06-1377

Manchado, E., Weissmueller, S., Morris, J. P., Chen, C.-C., Wullenkord, R., Lujambio, A., et al. (2016). A combinatorial strategy for treating kras-mutant lung cancer. *Nature* 534, 647–651. doi: 10.1038/nature18600

Martin, S., Zhang, Z., Martino, A., and Faulon, J.-L. (2007). Boolean dynamics of genetic regulatory networks inferred from microarray time series data. *Bioinformatics* 23, 866–874. doi: 10.1093/bioinformatics/btm021

Mendoza, M. C., Er, E. E., and Blenis, J. (2011). The Ras-ERK and PI3K-mTOR pathways: cross-talk and compensation. *Trends Biochem. Sci.* 36, 320–328. doi: 10.1016/j.tibs.2011.03.00

Ostrowski, M., Paulevé, L., Schaub, T., Siegel, A., and Guziolowski, C. (2016). Boolean network identification from perturbation time series data combining dynamics abstraction and logic programming. *Biosystems* 149, 139–153. doi: 10.1016/j.biosystems.2016.07.009

Poulikakos, P. I., Zhang, C., Bollag, G., Shokat, K. M., and Rosen, N. (2010). RAF inhibitors transactivate RAF dimers and ERK signalling in cells with wild-type BRAF. *Nature* 464:427. doi: 10.1038/nature08902

Roux, P. P., Ballif, B. A., Anjum, R., Gygi, S. P., and Blenis, J. (2004). Tumor-promoting phorbol esters and activated Ras inactivate the tuberous sclerosis tumor suppressor complex via p90 ribosomal S6 kinase. *Proc. Natl. Acad. Sci. U.S.A.* 101, 13489–13494. doi: 10.1073/pnas.0405659101

Saez-Rodriguez, J., Alexopoulos, L. G., Zhang, M., Morris, M. K., Lauffenburger, D. A., and Sorger, P. K. (2011). Comparing signaling networks between normal and transformed hepatocytes using discrete logical models. *Cancer Res.* 71, 5400–5411. doi: 10.1158/0008-5472.CAN-10-4453

Saez-Rodriguez, J., Simeoni, L., Lindquist, J. A., Hemenway, R., Bommhardt, U., Arndt, B., et al. (2007). A logical model provides insights into T cell receptor signaling. *PLoS Comput. Biol.* 3:e163. doi: 10.1371/journal.pcbi.0030163

Saini, K. S., Loi, S., de Azambuja, E., Metzger-Filho, O., Saini, M. L., et al. (2013). Targeting the PI3K/AKT/mTOR and Raf/MEK/ERK pathways in the treatment of breast cancer. *Cancer Treat. Rev.* 39, 935–946. doi: 10.1016/j.ctrv.2013.03.009

Shaw, R. J., and Cantley, L. C. (2006). Ras, PI(3)K and mTOR signalling controls tumour cell growth. *Nature* 441, 424–430. doi: 10.1038/nature04869

Streck, A. (2015). *Toolkit for Reverse Engineering of Molecular Pathways via Parameter Identification*. Ph.D. thesis, Freie Universität Berlin.

Streck, A., Thobe, K., and Siebert, H. (2015). "Analysing cell line specific EGFR signalling via optimized automata based model checking," in *Computational Methods in Systems Biology*, eds O. Roux and J. Bourdon (Cham: Springer), 264–276.

Streck, A., Thobe, K., and Siebert, H. (2016). Data-driven optimizations for model checking of multi-valued regulatory networks. *Biosystems* 149, 125–138. doi: 10.1016/j.biosystems.2016.05.004

Terfve, C., Cokelaer, T., Henriques, D., MacNamara, A., Goncalves, E., Morris, M. K., et al. (2012). CellNOptR: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms. *BMC Syst. Biol.* 6:133. doi: 10.1186/1752-0509-6-133

Thobe, K. (2017). *Logical Modeling of Uncertainty in Signaling Pathways of Cancer Systems*. Ph.D. thesis, Freie Universität Berlin.

Thobe, K., Sers, C., and Siebert, H. (2017). Unraveling the regulation of mTORC2 using logical modeling. *Cell Commun. Signal.* 15, 6–21. doi: 10.1186/s12964-016-0159-5

Thobe, K., Streck, A., Klarner, H., and Siebert, H. (2014). "Model integration and crosstalk analysis of logical regulatory networks," in *Computational Methods in Systems Biology*, eds P. Mendes, J. O. Dada, and K. Smallbone (Cham: Springer), 32–44.

Thomas, R. (1991). Regulatory networks seen as asynchronous automata: a logical description. *J. Theor. Biol.* 153, 1–23. doi: 10.1016/S0022-5193(05)80350-9

Videla, S., Saez-Rodriguez, J., Guziolowski, C., and Siegel, A. (2017). caspo: a toolbox for automated reasoning on the response of logical signaling networks families. *Bioinformatics* 33, 947–950. doi: 10.1093/bioinformatics/btw738

Wang, R.-S., Saadatpour, A., and Albert, R. (2012). Boolean modeling in systems biology: an overview of methodology and applications. *Phys. Biol.* 9:055001. doi: 10.1088/1478-3975/9/5/055001

Wilhelm, S. M., Carter, C., Tang, L., Wilkie, D., McNabola, A., Rong, H., et al. (2004). BAY 43-9006 exhibits broad spectrum oral antitumor activity and targets the RAF/MEK/ERK pathway and receptor tyrosine kinases involved in tumor progression and angiogenesis. *Cancer Res.* 64, 7099–7109. doi: 10.1158/0008-5472.CAN-04-1443

Will, M., Qin, A. C. R., Toy, W., Yao, Z., Rodrik-Outmezguine, V., Schneider, C., et al. (2014). Rapid induction of apoptosis by PI3K inhibitors is dependent upon their transient inhibition of RAS-ERK signaling. *Cancer Discov.* 4, 334–347. doi: 10.1158/2159-8290.CD-13-0611

Winter, J. N., Jefferson, L. S., and Kimball, S. R. (2011). ERK and Akt signaling pathways function through parallel mechanisms to promote mTORC1 signaling. *Am. J. Physiol.* 300, C1172–C1180. doi: 10.1152/ajpcell.00504.2010

Wong, K.-K., Engelman, J. A., and Cantley, L. C. (2010). Targeting the PI3K signaling pathway in cancer. *Curr. Opin. Genet. Dev.* 20, 87–90. doi: 10.1016/j.gde.2009.11.002

Yaktapour, N., Übelhart, R., Schüler, J., Aumann, K., Dierks, C., Burger, M., et al. (2013). Insulin-like growth factor-1 receptor (IGF1R) as a novel target in chronic lymphocytic leukemia. *Blood* 122, 1621–1633. doi: 10.1182/blood-2013-02-484386

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

TECHNOLOGY REPORT
published: 19 November 2018
doi: 10.3389/fphys.2018.01605

BioLQM: A Java Toolkit for the Manipulation and Conversion of Logical Qualitative Models of Biological Networks

*Aurélien Naldi**

Computational Systems Biology Team, Institut de Biologie de l'École Normale Supérieure, École Normale Supérieure, CNRS, INSERM, PSL Université, Paris, France

Here we introduce bioLQM, a new Java software toolkit for the conversion, modification, and analysis of Logical Qualitative Models of biological regulatory networks. BioLQM provides core modeling operations as building blocks for the development of integrated modeling software, or for the assembly of heterogeneous analysis workflows involving several complementary tools. Based on the definition of multi-valued logical models, bioLQM implements import and export facilities, notably for the recent SBML qual exchange format, as well as for formats used by several popular tools, facilitating the design of workflows combining these tools. Model modifications enable the definition of various perturbations, as well as model reduction, easing the analysis of large models. Another modification enables the study of multi-valued models with tools limited to the Boolean case. Finally, bioLQM provides a framework for the development of novel analysis tools. The current version implements various updating modes for model simulation (notably synchronous, asynchronous, and random asynchronous), as well as some static analysis features for the identification of attractors. The bioLQM software can be integrated into analysis workflows through command line and scripting interfaces. As a Java library, it further provides core data structures to the GINsim and EpiLog interactive tools, which supply graphical interfaces and additional analysis methods for cellular and multi-cellular qualitative models.

Keywords: qualitative modeling, computational systems biology, biological networks, boolean networks, static analysis, model conversion

OPEN ACCESS

Edited by:
Yoram Vodovotz,
University of Pittsburgh, United States

Reviewed by:
Noriko Hiroi,
Keio University, Japan
David McMillen,
University of Toronto Mississauga, Canada

***Correspondence:**
Aurélien Naldi
aurelien.naldi@ens.fr

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: *04 April 2018*
Accepted: *25 October 2018*
Published: *19 November 2018*

Citation:
Naldi A (2018) BioLQM: A Java Toolkit for the Manipulation and Conversion of Logical Qualitative Models of Biological Networks. Front. Physiol. 9:1605. doi: 10.3389/fphys.2018.01605

1. INTRODUCTION

Logical models are highly abstract dynamical models, which have been proposed to study biological regulatory systems in the late 60s (Kauffman, 1969; Thomas, 1973). This modeling framework has since gained popularity (Bornholdt, 2005; Saadatpour and Albert, 2013; Samaga and Klamt, 2013) and has been successfully applied to a wide range of regulatory and signaling systems (Saez-Rodriguez et al., 2007; Naldi et al., 2010; Helikar et al., 2013; Abou-Jaoudé et al., 2016).

In logical models, components are represented by discrete variables with a small range of possible values, representing qualitative differences in activity. Boolean components can only be active (1) or inactive (0), while multi-valued components define multiple activity levels. Regulatory effects are often represented as signed arcs between components in the **regulatory graph**. These effects are further formalized as logical rules (also called logical parameters or logical functions),

specifying the target activity level of each component according to the current levels of its regulators (a subset of all model components). Interactive software for model definition such as GINsim (Naldi et al., 2018a) or The Cell Collective (Helikar et al., 2012) enable the definition of regulatory graphs and logical rules. However, these logical rules are self-contained and can be used to recover signed regulatory interactions. The relative simplicity of this formalism enables the definition of large models with dozens of components, without requiring precise knowledge of kinetic parameters. A formal definition of logical qualitative models is provided in **Appendix 1** in Supplementary Material.

The CoLoMoTo consortium was recently founded to facilitate model sharing and foster cooperation in the qualitative modeling community, building on the introduction of the SBML qual exchange format (Chaouiya et al., 2013; Naldi et al., 2015). The bioLQM toolkit presented here reinforces this effort by implementing a collection of model modification, format conversion, and dynamical analysis operations in an extensible architecture illustrated in **Figure 1**. On one hand, format conversions enable the integration of several software tools in complex analysis workflows. On the other hand, the core data structure and model modifications provide building blocks for the development of integrated modeling tools, which can add their own model edition and visualization capabilities. BioLQM is notably embed in the popular GINsim software (Naldi et al., 2018a), which provides a graphical interface to most of its features. It is also used as backend for model definition and computation of successor states in Epilog (Varela et al., 2013), as well as in the CoLoMoTo notebook for model conversion and some dynamical analysis features (Naldi et al., 2018b). Preliminary versions of this toolkit were mentioned as the "LogicalModel" library Chaouiya et al. (2013) and Naldi et al. (2015).

Section 2 introduces model loading, saving and converting operations. Section 3 introduces the simulation and dynamical analysis features. Section 4 introduces model modifications. Section 5 illustrates the use of these features through the command-line and scripting interfaces for the analysis of a small model of the p53-Mdm2 network controlling DNA repair.

2. LOADING AND CONVERTING LOGICAL QUALITATIVE MODELS

The increasing use of qualitative models to study biological systems led to the development of various software tools for the logical formalism (Albert et al., 2008; Garg et al., 2008; Müssel et al., 2010; Terfve et al., 2012; Naldi et al., 2018a) and related qualitative approaches (Batt et al., 2012; Paulevé, 2017; Stoll et al., 2017). Most software tools use their own file format for the definition of models, hindering the delineation of analysis workflows combining different tools. The SBML qual exchange format (Chaouiya et al., 2013) has recently been proposed to improve interoperability between modeling tools. However SBML support is often missing from existing software and may not be a priority for newer ones.

To ease model exchange between software tools that do not all support the SBML qual format, the bioLQM toolkit provides an extensible list of format handlers connected to the internal model representation. Each format is described as a Java class providing annotations (name of the format, default file extension and multi-valued support) along with optional implementations of model import (loading a file into the internal representation) or export (saving the internal representation to a file) operations. These descriptor classes are available through service discovery to facilitate the addition of new formats.

The supported formats are listed in **Table 1** and in bioLQM documentation [1]. BioLQM uses JSBML (Rodriguez et al., 2015) to load and save SBML qual models. The other import parsers are based on the antlr parser generator (Parr and Quong, 1995). While some formats natively support multi-valued models, many are limited to the Boolean case. Multi-valued models can be exported to these Boolean formats through an implicit booleanization step, described in section 4.

3. MODEL DYNAMICS AND SIMULATION

A **state** of a model is a vector giving the activity levels of all its components. As the activity level of each component is restricted to a finite range, the **state space** (containing all possible states) itself is also finite. However, the total number of possible states grows exponentially with the number of components. We say that a component is **called to update** in a given state if the evaluation of the associated logical rule is different from its current activity level: for example an inactive component can become active. **Stable states** (also called fixed points, or steady states) are states in which no component is called to update. Such stable states denote a qualitative equilibrium in which all components can maintain their current activity level.

The dynamics of the model (i.e., its evolution over time) is given by transitions between states of the model, controlled by the updating calls (i.e., by the logical rules of the model) and by **updating modes** which define the synchronization between concurrent updating calls. Various types of updating modes have been introduced, with most software tools focusing on a specific subset. BioLQM aims to provide an extensive choice of updating modes in a single toolkit. In the following subsections, we further distinguish deterministic and non-deterministic simulations and provide an overview of all updating modes implemented in bioLQM. While stable states, which have no transition toward other states, do not depend on the updating mode, reachability properties and cyclical attractors can be strongly affected by the choice of updating mode as illustrated in **Figure 2**. More formal definitions of the updating calls and updating modes are given in Appendix 2 in **Supplementary Material**.

3.1. Deterministic Simulations

In a **deterministic** simulation, each state has a unique successor, except stable states which have no successor at all as we consider here that a successor must denote a change of state. Starting with an initial state, a deterministic simulation yields an ordered list

[1] See http://colomoto.org/biolqm/

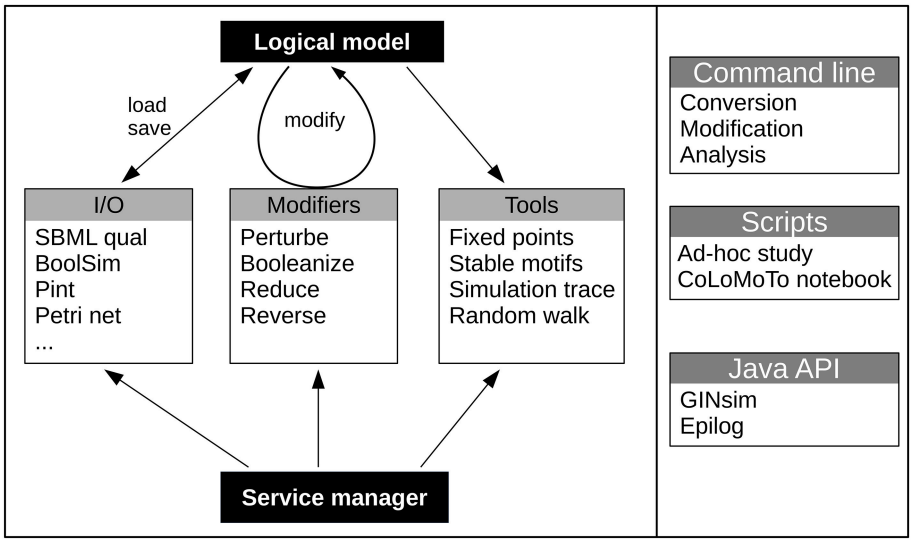


FIGURE 1 | Global structure of the bioLQM toolkit. The bioLQM toolkit is centered around a data structure for the representation of logical qualitative models. Based on this data structure, (i) the `I/O` module contains a collection of formats enabling model loading and saving ; (ii) the `modifiers` module contains a collection of model modifiers to transform an input model into a modified model ; (iii) the `tools` module contains a collection of analysis tools. All these feature are accessible through a central `service manager`, which handles service discovery and serves as main entry point for the Java API. A simple `command line launcher` provides quick execution of simple workflows, while a `scripting engine` can be used for more complex use cases.

TABLE 1 | Available formats.

File extension	Multi-valued	Import	Export	Description and associated tools
sbml	x	x	x	SBML qual Exchange format (Chaouiya et al., 2013)
bnet		x	x	(Py)BoolNet (Müssel et al., 2010; Klarner et al., 2017)
booleannet		x	x	booleannet (Albert et al., 2008)
boolfunction		x	x	Boolean functions
boolsim		x	x	boolsim (genYsis) (Garg et al., 2008)
cnet		x	x	BNS (Dubrova and Teslenko, 2011)
ginml	x		x	GINsim (Naldi et al., 2018a)
mnet	x	x	x	Custom text format for multi-valued models
tt	x	x	x	Truth table
an	x		x	Pint automata network (Paulevé, 2017)
apnn, pnml, ina	x		x	Conversion to Petri Net formats (Chaouiya et al., 2011)
gna	x		x	GNA (Piecewise-linear formalism) (Batt et al., 2012)
bnd			x	MaBoSS (Stochastic Boolean model) (Stoll et al., 2017)

The Import/Export capabilities are listed in the corresponding columns (all formats can be exported). The formats natively supporting multi-valued models are also identified, other formats rely on implicit model booleanization.

of successive states, called a **trace**. Given a sufficient number of steps, all traces end in an **attractor**, which can be either a stable state or a **cyclical attractor** of length k in which the k-th successor of each state is itself. The `trace` tool, illustrated in section 5, uses an initial state and a deterministic updater to compute a simulation trace. The following deterministic updating modes are supported:

- The **synchronous** (or parallel) updating applies all logical rules at the same time (Kauffman, 1969).
- The **sequential** updating applies all rules in a predetermined order. Instead of evaluating all rules on the original state before updating all components at once as in the synchronous case, they are evaluated on the state obtained after applying the previous rule. The selected order can then change dramatically the successor state: a different sequential updater can be defined for each possible ordering.
- The **block-sequential** updating generalizes the sequential one by considering groups of components updated synchronously

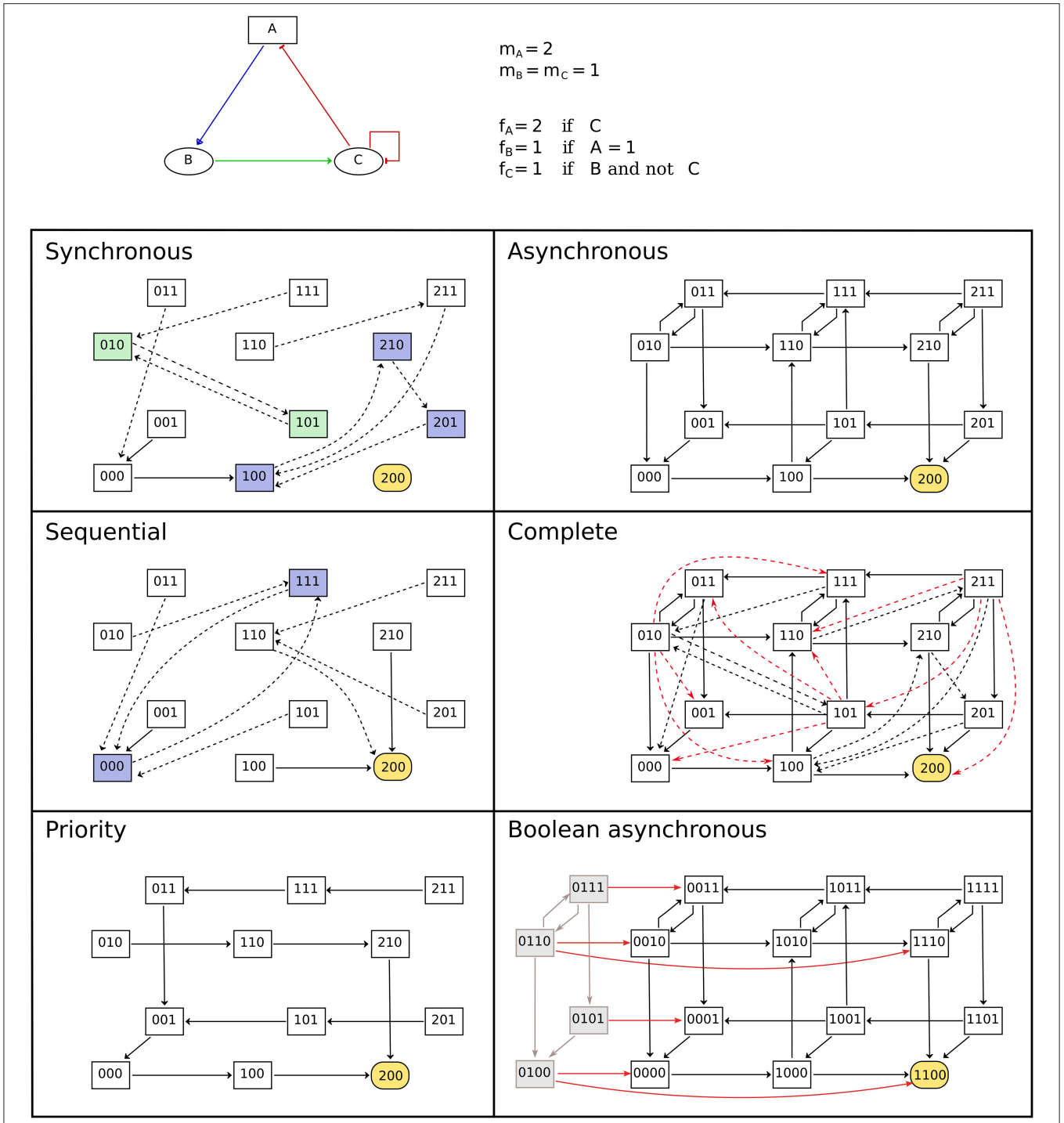


FIGURE 2 | Comparison of updating modes. State transition graphs obtained with the multi-valued model shown in the top part using various deterministic (left-side) and non-deterministic (right-side) updatings. Dashed arcs denote multiple transitions and node coloring emphasizes attractors. Note that the stable state is common to all updating modes. The sequential and priority updaters follow the implicit ordering of the components. The STG obtained with the complete updating contains all synchronous and asynchronous transitions, as well as additional transitions to leave the states encompassing more than two updating calls. These transitions are colored in red in the corresponding panel. Finally, the bottom-right panel contains the asynchronous STG obtained for the booleanized version of the model. In this STG, gray nodes and arcs on the left side correspond to non-admissible states and transitions between them. These states are unreachable from the admissible ones, and the transitions enabling to leave this set of states are highlighted.

(Robert, 1986). The definition of a block-sequential updater relies on an ordered partition of the model components.
- The **synchronous priority** updating is also based on a partition of components into blocks, but only the first block containing updated components will be considered. The set of possible updaters is a subset of the priority-based updaters introduced by Fauré et al. (2006).

3.2. Non-deterministic Simulations

In a **non-deterministic** simulation, each state can have several successors. Starting with an initial state, a non-deterministic simulation can lead to a large number of alternative trajectories. This type of dynamics if often represented as a **State Transition Graph** (STG), where the nodes are states of the model, and arcs denote possible transitions between these states. Like in the deterministic case, all trajectories end in an attractor, but starting from an initial state, a non-deterministic simulation can lead to several alternative attractors. These attractors can be **stable states**, **cyclical attractors**, as well as sets of intertwined cycles called **complex attractors**. More formally, all attractors are terminal strongly connected components of the STG. State transition graphs can represent deterministic traces as well as more complex dynamical behaviors. Such a graph can cover several alternative initial states or even all possible states. The current version of bioLQM supports the definition of non-deterministic updaters, enabling the computation of the lists of successor states. However, it does not provide a complete engine for non-deterministic simulations, or a data structure for state transition graphs. GINsim (Naldi et al., 2018a) implements these features on top of bioLQM. The following non-deterministic updating modes are supported:

- The **asynchronous** updating applies all logical rules independently. All successors of a state change exactly one component (Thomas, 1973).
- The **complete** updating considers all possible combination of components to be updated at once. The set of successors includes all asynchronous successors, as well as the synchronous one (and more).
- The **priority** updating generalizes the *synchronous priority* introduced above by allowing some of the blocks (priority classes) to be updated asynchronously (Fauré et al., 2006).

3.3. Stochastic Simulations

Stochastic updaters enable the computation of a single successor, which is selected randomly among multiple possibilities and can thus change between calls. A stochastic updater can be derived from any non-deterministic updater by assigning identical probabilities to all transitions defined by the original updater. Alternatively, a custom updater can be constructed by defining individual probabilities.

BioLQM provides the `random` tool to compute single random trajectories using the above stochastic updaters. This tool is limited to the construction of individual trajectories and does not provide a complete stochastic analysis. As listed in **Table 1**, bioLQM enables the conversion of Boolean models to the format of the MaBoSS software, which uses the Gillespie algorithm to estimate the probabilities of Boolean states of a continuous time Markov process, and provides a collection of scripts to further analyze the simulation results (Stoll et al., 2017).

3.4. Identification of Attractors

The dynamical analysis of large regulatory networks through model simulation suffers from combinatorial explosion, especially in the non-deterministic case. BioLQM implements two published methods based on constraint-solving for the identification of attractors without explicit state enumeration.

1. The first method enables the identification of **stable states** (fixed points) by extracting and combining stability conditions from the logical rules (Naldi et al., 2007). BioLQM includes this implementation, using decision diagrams to manipulate stability conditions, and introduces an alternative implementation based on the clingo ASP solver (Gebser et al., 2011), which tends to be slower for small models, but can scale better in some cases. Similar methods are also available in the GNA and Pint tools (Batt et al., 2012; Paulevé, 2017).
2. The efficient identification of cyclical attractors and complex attractors remain a challenging problem, especially as these attractors can depend on the updating mode. **Stable patterns** have recently been proposed as an approximation of complex attractors, which can be identified efficiently and does not depend on the updating mode (Zañudo and Albert, 2013; Klarner et al., 2014). Here, a pattern is a partially-defined state where some components have a fixed activity level, while others are undefined. Such a pattern represents all states with matching activity levels for the defined components (i.e., 2^k possible states for k undefined Boolean components). A pattern is stable if the images of all included states belong to the pattern (the image of a state is its successor in a synchronous updating). BioLQM proposes an adapted version of the method implemented in PyBoolNet (Klarner et al., 2014, 2017) using the clingo ASP solver (Gebser et al., 2011), and introduces a new alternative implementation based on decision diagrams.

While complex attractors are well estimated through stable patterns, their exact identification requires further analysis using external software tools, adapted to the selected updating mode. In the synchronous case, the BNS tool (Dubrova and Teslenko, 2011) identifies cyclical attractors of length k using constraint solving. This approach could be extended to other deterministic updatings, but can not handle non-deterministic cases. In contrast, BoolSim uses symbolic exploration for the identification of complex attractors in the synchronous and asynchronous case (Garg et al., 2008). While this approach scales better than simple simulation, it is more sensitive to combinatorial explosion than approaches based on constraint-solving. To perform the analysis provided by the BoolSim and BNS tools, bioLQM can convert models to their respective formats.

4. MODEL MODIFICATIONS

Several software tools propose to emulate biological **mutations** by constructing model variants in which one or several logical

rules have been modified. In bioLQM, the various **model modification** tools enable the flexible definition of model variants. The resulting modified models can have a different set of components than the original model. Each modification can be described by a keyword (identifier of the type of modification) and some parameters. The model modifier API in bioLQM allows to chain several modifications before model conversion or analysis. The following describes the various types of model modifications implemented in bioLQM.

Perturbations

A **perturbation** (often called **mutation**) enables to change some of the logical rules of a model. BioLQM provides three types of "atomic perturbations" (fixed value, range restriction, and removal of a regulator) which modify a single logical rule. They are briefly described below, more formal definitions can be found in Appendix 3 in **Supplementary Material**. "Multiple perturbations" can then be used to combine several atomic perturbations. The definition of these perturbations is supported by a simple syntax, as illustrated in section 5.3 and described in the online documentation.

Perturbations are commonly used to model gene knockouts by **fixing the activity level** of the corresponding component to 0, or ectopic expressions by fixing it to 1. Multi-valued components can also be fixed to a higher activity level (inside their normal activity range).

Restricting the activity range of multi-valued components enables to account for a partially impaired activity ([loss of the higher activity level(s)] or to set a minimal activity level.

Lastly, it is possible to define the **perturbation of a single interaction**, i.e., to remove one of the regulators of a component. This type of perturbation enables for example the definition of the loss of a single binding site preventing the action of the source component on a subset of its targets. The removal of an interaction amounts to rewrite the logical rule of its target component. Note that the atomic perturbation describes the effect on a single target: a single "biological mutation" may correspond to a "multiple perturbation" in the model if several targets are affected by the loss of the same binding site. This type of perturbation is also convenient to evaluate the importance of an interaction representing an hypothetical effect.

4.1. Model Reduction

Model reduction aims to ease the analysis of models with a large number of components by constructing a smaller model involving fewer components, but exhibiting similar dynamical properties. BioLQM provides a model reduction method which updates the logical rules of the remaining components to emulate the effect of the removed components (Naldi et al., 2011; Veliz-Cuba, 2011). This reduction preserves key dynamical properties of the model, in particular the stable states and stable patterns. However, it can affect some dynamical properties, depending on the choice of reduced components.

This modifier usually relies on the specification of the set of components to reduce. Some types of reduction can be fully automated. In particular, bioLQM supports the reduction of output components, which was shown to preserve attractors and reachability properties (Naldi et al., 2012), as well as the propagation of fixed components, which has also been shown to preserve attractors (Saadatpour et al., 2013).

After reduction, the reduced components are not fully eliminated from bioLQM: they are no longer allowed to regulate other components, but they keep a logical rule to allow the computation of their expected value in the reduced model.

4.2. Boolean Mapping of Multi-Valued Models

As discussed above, some software tools and formats are limited to Boolean models, for example as they rely on specific theoretical results or data structures. To apply such software tools to the analysis of a multi-valued model, we can construct a Boolean model such that its dynamical properties can be transferred to the original multi-valued model.

This **model Booleanization** step is based on the Boolean mapping discussed by Didier et al. (2011). In this mapping, a multi-valued component with a maximal activity level m is replaced by m Boolean components, each denoting increasing activity. All possible states of the original model can then be associated to states of the Boolean model. The logical rules of the new model ensure that we obtain the same transitions between these states. However, some states of the Boolean model are not mapped to states of the original model. These additional states are called "non-admissible states."

The dynamical properties observed on the admissible states of the Boolean model can be transferred to the original model. The implementation proposed here further ensures that all synchronous and asynchronous simulations starting with a non-admissible state can lead to an admissible state after a sufficient number of steps. This property ensures that no attractor contains any non-admissible state (see **Figure 2**).

Model Booleanization is used automatically when converting multi-valued models to formats supporting only Boolean models. It can also be performed explicitly, like other model modifications.

5. USE CASE: ANALYSIS OF THE P53-MDM2 NETWORK

The cellular response to DNA damage relies on the p53 transcription factor, which induces the synthesis of DNA repair proteins. The ubiquitin ligase Mdm2 blocks the transcriptional activity on p53 in the nucleus, while p53 activates the transcription of Mdm2 and inhibits its nuclear translocation. In this section, we use a logical model involving DNA damage, p53, Mdm2 in the cytoplasm and Mdm2 in the nucleus. See the recently published GINsim tutorial Naldi et al. (2018a) and the enclosed references for a more complete description of this system and its encoding into a logical model.

In the following, we define the model in a text file named `p53.mnet`, using a simple text format for the definition of multi-valued logical models (p53 and cytoplasmic Mdm2 are represented by ternary components). Each line of the file reproduced below assigns a logical function to one of the

components of the model. The line starts with the the identifier of the component, separated from the function itself by a leftwards arrow (`<-`). The `&`, `|`, and `!` symbols stand for the `AND`, `OR`, and `NOT` operations respectively. The colon character (`:`) is used to specify multi-valued thresholds, both for assigning the target component and inside the functions.

```
DNAdam    <- DNAdam & !p53:2
p53:2     <- !Mdm2nuc
Mdm2cyt:1 <- !p53:2
Mdm2cyt:2 <- p53:2
Mdm2nuc   <- Mdm2cyt:2 | (Mdm2cyt & !p53 & !DNAdam)
```

5.1. Install and Launch bioLQM

Documentation, source code and releases (under the LGPL v3 license) are available on http://colomoto.org/biolqm. BioLQM is distributed as a JAR file[2], which can be launched with the command `java -jar bioLQM.jar`. In this section, we will use the `bioLQM` command as shorthand.

5.2. Resting State and DNA Repair

We start by looking for the stable states (fixed points) of this model. For this, we launch bioLQM on the command-line, load the model from the `p53.mnet` file defined above, and run the `fixpoints` tool. The corresponding command line and its output are reproduced hereafter.

```
$ bioLQM p53.mnet -r fixpoints
DNAdam p53 Mdm2cyt Mdm2nuc
0011
```

In the output, bioLQM displays a first line with the list of components, followed by a line for each identified stable state, giving the activity level of each component in the same order. The p53-Mdm2 model has a single stable state corresponding to a resting state in absence of DNA damage. In this state, the basal activity of Mdm2 prevents p53 activation. This analysis shows all the stable states of the model, but does not identify more complex attractors. We can then use the trapspace tool to identify stable patterns, which provide a good approximation of complex attractors in practice.

```
$ bioLQM p53.mnet -r trapspace
DNAdam p53_b1 p53_b2 Mdm2cyt_b1 Mdm2cyt_b2 Mdm2nuc
0 0 0 1 0 1
```

In this output, the two multi-valued components of the model have been extended to four Boolean components. While this requires a careful interpretation, it provides fine-grained results for complex attractors in which multi-valued components can be restricted to a range of their possible activity levels. Here we obtain a single pattern corresponding to the previously identified stable state. Note that this result does not strictly rule out the existence of a complex attractor, but attractors which do not correspond to such stable patterns are rare in practice and often depend on subtle delay effects. In this model, the resting state is indeed the only attractor.

We can then evaluate the behavior of this network upon addition of DNA damage to this resting state. For this, we use the `trace` tool to perform a synchronous simulation, starting with an initial state (defined after the `-i` flag) obtained by adding DNA damage to the resting state.

```
$ bioLQM p53.mnet -r trace -i 1011
1011
1010
1110
1210
0220
0221
0121
0011
```

In this simulation trace, we see that the introduction of DNA damage in the resting state leads to the inactivation of Mdm2 in the nucleus, enabling the activation of p53. This triggers DNA repair and allows Mdm2 to accumulate in the cytoplasm. Finally, Mdm2 can enter the nucleus and inhibit p53, coming back to the resting state. In this simulation, we assume that all possible transitions happen synchronously in each state, which could lead to artefactual trajectories. Asynchronous simulations are widely considered as more reliable, but they lead to a large number of alternative branches and are not well suited for simple command-line simulations. We can however perform a random walk in the set of possible asynchronous trajectories using the `random` tool. In this case, all asynchronous trajectories eventually lead to the same stable state (not illustrated here).

5.3. Definition of Model Perturbation

We then apply a perturbation to study the impact of a p53 knockout on the list of stable states. The `-m perturbation` parameters trigger the construction of a modified model. The following parameters (up to the next flag starting with a minus sign) define the modified functions. Here `p53%0` describes a loss of p53 activity.

```
$ bioLQM p53.mnet -m perturbation p53%0 -r fixpoints
DNAdam p53 Mdm2cyt Mdm2nuc
0011
1010
```

We see that the resting state is still valid in the p53 knockout, however a new stable state appears in which DNA damage could not be repaired.

Instead of a full knockout of p53, we then evaluate a more subtle perturbation in which only its ability to trigger the DNA repair machinery is impaired. This corresponds to the removal of the interaction between p53 and DNAdam in our model.

```
$ bioLQM p53.mnet -m perturbation p53:DNAdam%0 -r fixpoints
DNAdam p53 Mdm2cyt Mdm2nuc
0011
```

```
$ bioLQM p53.mnet -m perturbation p53:DNAdam%0 -r trapspace
DNAdam p53_b1 p53_b2 Mdm2cyt_b1 Mdm2cyt_b2 Mdm2nuc
0 0 0 1 0 1
1 - - 1 - -
```

Here we see that this perturbation does not affect the stability of the resting state, and does not create an additional stable state as in the full p53 knockout case. However, the trapspace tool reveals the creation of a complex attractor involving oscillations

[2]It requires a Java Runtime Environment, see https://www.java.com

of p53 and Mdm2. Note that these oscillations exist transiently in the original model but lead back to the resting state after DNA repair.

5.4. Model Conversion Enables Interoperability

As discussed in section 2, the analysis of complex models can combine several software tools. After running the following command, the new `p53.sbml` file will contain the functions defined above in the `SBML qual` format. This format is suitable for use in several other tools, or for submission in the BioModels database (Chelliah et al., 2013).

```
$ bioLQM p53.mnet p53.sbml
```

5.5. Definition of Complex Analysis as Scripts

More complex analysis tasks can use the integrated **scripting interface**. Based on the java scripting engine, it supports scripts written in javascript (as part of the java platform) or in another supported language by providing additional libraries (including python and lua). The following sample script generates all possible individual knockout perturbations, and saves each modified model.

```
filename = lqm.args[0]
model = lqm.load(filename)
nodes = model.getComponents()
for (i in nodes) {
    node = nodes[i]
    perturbed = lqm.modify(model, 'perturbation', node+'%0')
    lqm.save(perturbed, filename+"_"+node+"_KO.mnet")
}
```

This script can be launched using the `-s` flag, followed by the script file name. Additional arguments can be used to adapt the behavior of the script. In this example, we specify the name of the original model file.

```
$ bioLQM -s generate_perturbations.js p53.mnet
```

The recently introduced CoLoMoTo Docker image (Naldi et al., 2018b) provides a python API integrating several complementary software tools. This environment includes a dedicated python API for bioLQM, which plays a central role in model conversion.

6. SUMMARY AND DISCUSSION

The increasing use of logical models of biological regulatory networks led to the development of multiple complementary software tools for their analysis. The recent introduction of the SBML qual format (Chaouiya et al., 2013) and the formation of the CoLoMoTo consortium (Naldi et al., 2015) aims to facilitate the exchange of models between tools. The bioLQM toolkit enables the use of additional software tools through conversion to their native formats. It provides model conversion operations in the CoLoMoTo notebook (Naldi et al., 2018b), enabling the delineation of analysis workflow involving a series of different tools.

BioLQM can also be used to apply various perturbations to the converted models, enabling the study of model variants emulating a knockout, an ectopic activity, or the loss of an interaction. Model modifications include the booleanization of multi-valued models for analysis with tools restricted to a Boolean formalism, as well as model reduction, decreasing the number of components to ease the analysis of complex models.

Finally, bioLQM provides several internal tools for the dynamical analysis of logical models. Two of the included tools allow the construction of deterministic and stochastic simulation traces, based on a comprehensive collection of updating modes. BioLQM also implements non-deterministic updating modes, which can be used as core components of complete simulation engines, as done by the GINsim (Naldi et al., 2018a) and Epilog (Varela et al., 2013) software suites. Two other tools enable the efficient identification of stable states and the approximation of most complex attractors.

The features described above are organized in a flexible architecture to facilitate the addition of new modules (file formats, model modifications, analysis tools) and to provide a consistent API. In the next version, the configuration API of analysis tools will be further improved to improve their use through python scripts in the new CoLoMoTo notebook.

Hardware requirements strongly depend on the size and structure of models and the operations performed. The complexity of individual logical rules can be a limiting factor: components with tens of regulators could have intricate rules with high computational cost. Fortunately, such rules are seldom used in biological models. Any desktop computer should be able to load and convert most models, including large ones. However, detailed dynamical analysis of models beyond 30 components can rapidly fill the available memory. In bioLQM, the `fixpoints` and `trapspace` analysis tools rely on efficient constraint-solving methods, which can scale to hundreds of components. The `trace` and `random` simulation tools are designed to work on large models as well by avoiding to store all visited states and interrupting the simulation when a stable state is reached or after a limit on the number of steps. In future versions, these tools will further use the identified trapspaces to interrupt the simulation when reaching a complex attractor.

BioLQM uses decision diagrams to store the logical rules internally, which enforces a normalized representation of the function, depending on the ordering of components. It has the advantage of providing guarantees on the number of tests to perform to evaluate a function, but it replaces the original representation of the function, making it harder to manipulate by the user afterwards. Future version will include several alternative representations to preserve hand-crafted logical functions through conversion (when the output format allows it).

Logical models are non-deterministic when using the asynchronous updating, however individual logical rules are deterministic: they associate a single "target value" to each state of the system. The ability to lift this limitation is considered in the design of the new internal data structure, but is not an immediate goal: the next releases of bioLQM will remain focused on deterministic functions.

In logical models (and by extension in bioLQM), each component is associated to its own logical rule, however the Petri net and automata network formalisms separate components from transitions. This separation allows in particular the definition of transitions affecting several components simultaneously. Such behaviors could be emulated in logical models through the addition of synchronizing components. Proper support for this use case would require extensions of the SBML qual specification, as well as changes in the internal data structure.

Like most modeling tools, bioLQM is currently centered on logical rules, however a complete model may contain important additional information, such as annotations and graphical layout information. Model annotations are supported in SBML core (without additional extensions), however annotations can be defined in any format, hindering interoperability. Further discussions are needed within the community to delineate best practices and ensure that annotations can be shared efficiently. Graphical layout information can be stored along with SBML qual models using a dedicated extension. This information is currently supported by JSBML and GINsim, it will be integrated in future versions of bioLQM. JSON "sidecar" files could then be used to facilitate the integration of such additional information with file formats which do not support it directly.

The reproducibility of model analysis relies on sharing both the model itself and the definition of simulation parameters, in particular initial states and updating modes. A single initial state can be defined in the SBML qual file. Additional initial states and simulation parameters fall in the scope of the Simulation Experiment Description Markup Language (SED-ML) format (Bergmann et al., 2015), which does not yet support qualitative models. Ongoing discussions should lead to extensions of the SED-ML format and the Kinetic Simulation Algorithm Ontology (Courtot et al., 2014) to describe model modifications and simulation parameters. These extensions will then be integrated into bioLQM and other qualitative modeling software.

AUTHOR CONTRIBUTIONS

The author confirms being the sole contributor of this work and has approved it for publication.

FUNDING

AN acknowledges support from the French Agence Nationale de la Recherche (ANR), in the context of the project SCAPIN [ANR-15-CE15-0006-01].

ACKNOWLEDGMENTS

This work benefited from the feedback of members of the CoLoMoTo consortium (colomoto.org). Pedro Monteiro, Claudine Chaouiya and Denis Theffry provided feedback for the integration in GINsim and Epilog. Julien Dorier and Gautier Stoll helped with the BoolSim and MaBoSS formats respectively. Loic Paulevé provided feedback and implemented the Pint format. Francisco Plana implemented the block-sequential updater. Hannes Klarner implemented the CNET and bnet formats. Céline Hernandez tested the API and provided feedback.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphys.2018.01605/full#supplementary-material

REFERENCES

Abou-Jaoudé, W., Traynard, P., Monteiro, P. T., Saez-Rodriguez, J., Helikar, T., Thieffry, D., et al. (2016). Logical modeling and dynamical analysis of cellular networks. *Front. Genet.* 7:94. doi: 10.3389/fgene.2016.00094

Albert, I., Thakar, J., Li, S., Zhang, R., and Albert, R. (2008). Boolean network simulations for life scientists. *Source Code Biol. Med.* 3:16. doi: 10.1186/1751-0473-3-16

Batt, G., Besson, B., Ciron, P., de Jong, H., Dumas, E., Geiselmann, J., et al. (2012). Genetic network analyzer: a tool for the qualitative modeling and simulation of bacterial regulatory networks. *Methods Mol. Biol.* 804, 439–462. doi: 10.1007/978-1-61779-361-5_22

Bergmann, F. T., Cooper, J., Le Novère, N., Nickerson, D., and Waltemath, D. (2015). Simulation Experiment Description Markup Language (SED-ML) Level 1 Version 2. *J. Integr. Bioinform.* 12:262. doi: 10.1515/jib-2015-262

Bornholdt, S. (2005). Systems biology. less is more in modeling large genetic networks. *Science* 310, 449–451. doi: 10.1126/science.1119959

Chaouiya, C., Bérenguier, D., Keating, S., Naldi, A., van Iersel, M., Rodriguez, N., et al. (2013). SBML qualitative models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools. *BMC Syst. Biol.* 7:135. doi: 10.1186/1752-0509-7-135

Chaouiya, C., Naldi, A., Remy, E., and Thieffry, D. (2011). Petri net representation of multi-valued logical regulatory graphs. *Nat. Comput.* 10, 727–750. doi: 10.1007/s11047-010-9178-0

Chelliah, V., Laibe, C., Le Novére, N. (2013). BioModels database: a repository of mathematical models of biological processes. *Methods Mol. Biol.* 1021, 189–199. doi: 10.1007/978-1-62703-450-0_10

Courtot, M., Juty, N., Knupfer, C., Waltemath, D., Zhukova, A., Drager, A., et al. (2014). Controlled vocabularies and semantics in systems biology. *Mol. Syst. Biol.* 7, 543–543. doi: 10.1038/msb.2011.77

Didier, G., Remy, E., and Chaouiya, C. (2011). Mapping multivalued onto boolean dynamics. *J. Theor. Biol.* 270, 177–184. doi: 10.1016/j.jtbi.2010.09.017

Dubrova, E., and Teslenko, M. (2011). A SAT-Based algorithm for finding attractors in synchronous boolean networks. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 8, 1393–1399. doi: 10.1109/TCBB.2010.20

Fauré, A., Naldi, A., Chaouiya, C., and Thieffry, D. (2006). Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle. *Bioinformatics* 22, 124–131. doi: 10.1093/bioinformatics/btl210

Garg, A., Cara, A., Xenarios, I., Mendoza, L., and Micheli, G. (2008). Synchronous versus asynchronous modeling of gene regulatory networks. *Bioinformatics* 24, 1917–1925. doi: 10.1093/bioinformatics/btn336

Gebser, M., Kaufmann, B., Kaminski, R., Ostrowski, M., Schaub, T., and Schneider, M. (2011). Potassco: the potsdam answer set solving collection. *AI Commun.* 24, 107–124. doi: 10.3233/AIC-2011-0491

Helikar, T., Kochi, N., Kowal, B., Dimri, M., Naramura, M., Raja, S., et al. (2013). A comprehensive, multi-scale dynamical model of ErbB receptor signal transduction in human mammary epithelial cells. *PLoS ONE* 8:e61757. doi: 10.1371/journal.pone.0061757

Helikar, T., Kowal, B., McClenathan, S., Bruckner, M., Rowley, T., Madrahimov, A. et al. (2012). The Cell Collective: toward an open and collaborative approach to systems biology. *BMC Syst. Biol.* 6:96. doi: 10.1186/1752-0509-6-96

Kauffman, S. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.* 22, 437–467. doi: 10.1016/0022-5193(69)90015-0

Klarner, H., Bockmayr, A., and Siebert, H. (2014). "Computing symbolic steady states of boolean networks," in *Cellular Automata*, Vol. 8751 of *Lect Notes Computing Science* (Krakow), 561–70.

Klarner, H., Streck, A., and Siebert, H. (2017). PyBoolNet: a python package for the generation, analysis and visualization of boolean networks. *Bioinformatics* 33, 770–772. doi: 10.1093/bioinformatics/btw682

Müssel, C., Hopfensitz, M., and Kestler, H. (2010). Boolnet–an r package for generation, reconstruction and analysis of boolean networks. *Bioinformatics* 26, 1378–1380. doi: 10.1093/bioinformatics/btq124

Naldi, A., Carneiro, J., Chaouiya, C., and Thieffry, D. (2010). Diversity and plasticity of th cell types predicted from regulatory network modelling. *PLoS Comput. Biol.* 6:e1000912. doi: 10.1371/journal.pcbi.1000912

Naldi, A., Hernandez, C., Abou-Jaoudé, W., Monteiro, P. T., Chaouiya, C., and Thieffry, D. (2018a). Logical modelling and analysis of cellular regulatory networks with GINsim 3.0. *Front. Physiol.* 7605:646. doi: 10.3389/fphys.2018.00646

Naldi, A., Hernandez, C., Levy, N., Stoll, G., Monteiro, P. T., Chaouiya, C., et al. (2018b). The CoLoMoTo interactive notebook: accessible and reproducible computational analyses for qualitative biological networks. *Front. Physiol.* 9:680. doi: 10.3389/fphys.2018.00680

Naldi, A., Monteiro, P., and Chaouiya, C. (2012). "Efficient handling of large signalling-regulatory networks by focusing on their core control," in *Computational Methods for Systems Biology*, Lecture Notes in Computer Science, eds D. Gilbert and M. Heiner (Berlin;Heidelberg: Springer), 288–306.

Naldi, A., Monteiro, P., Müssel, C., Consortium for Logical Models and Tools, Kestler, H., Thieffry, D., et al. (2015). Cooperative development of logical modelling standards and tools with colomoto. *Bioinformatics* 31, 1154–1159. doi: 10.1093/bioinformatics/btv013

Naldi, A., Remy, E., Thieffry, D., and Chaouiya, C. (2011). Dynamically consistent reduction of logical regulatory graphs. *Theor. Comput. Sci.* 412, 2207–2218. doi: 10.1016/j.tcs.2010.10.021

Naldi, A., Thieffry, D., and Chaouiya, C. (2007). "Decision diagrams for the representation and analysis of logical models of genetic networks," in *Computational Methods for Systems Biology*, volume 4695 of *Lecture Notes in Computer Science*, eds M. Calder and S. Gilmore (Berlin;Heidelberg: Springer), 233–247.

Parr, T. J., and Quong, R. W. (1995). ANTLR: A predicated-LL(k) parser generator. *Softw. Practice Exper.* 25, 789–810. doi: 10.1002/spe.4380250705

Paulevé, L. (2017). "Pint: a static analyzer for transient dynamics of qualitative networks with IPython interface," in *Computational Methods for Systems Biology*, Vol. 10545 of *Lecture Notes in Computer Science,* eds J. Feret and H. Koeppl (Cham: Springer), 370–316.

Robert, F. (1986). *Discrete Iterations : A Metric Study*. Springer. doi: 10.1007/978-3-642-61607-5

Rodriguez, N., Thomas, A., Watanabe, L., Vazirabad, I., Kofia, V., Gómez, H., et al. (2015). Jsbml 1.0: providing a smorgasbord of options to encode systems biology models. *Bioinformatics* 31, 3383–3386. doi: 10.1093/bioinformatics/btv341

Saadatpour, A., and Albert, R. (2013). Boolean modeling of biological regulatory networks: a methodology tutorial. *Methods* 62, 3–12. doi: 10.1016/j.ymeth.2012.10.012

Saadatpour, A., Albert, R., and Reluga, T. C. (2013). A Reduction Method for Boolean Network Models Proven to Conserve Attractors. *SIAM J. Appl. Dyn. Syst.* 12, 1997–2011. doi: 10.1137/13090537X

Saez-Rodriguez, J., Simeoni, L., Lindquist, J., Hemenway, R., Bommhardt, U., Arndt, B., et al. (2007). A logical model provides insights into T cell receptor signaling. *PLoS Comput. Biol.* 3:e163. doi: 10.1371/journal.pcbi.0030163

Samaga, R., and Klamt, S. (2013). Modeling approaches for qualitative and semi-quantitative analysis of cellular signaling networks. *Cell Commun. Signal* 11:43. doi: 10.1186/1478-811X-11-43

Stoll, G., Caron, B., Viara, E., Dugourd, A., Zinovyev, A., Naldi, A., et al. (2017). MaBoSS 2.0: an environment for stochastic Boolean modeling. *Bioinformatics* 33, 2226–2228. doi: 10.1093/bioinformatics/btx123

Terfve, C., Cokelaer, T., Henriques, D., MacNamara, A., Goncalves, E., Morris, M. K., et al. (2012). CellNOptR: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms. *BMC Syst. Biol.* 6:133. doi: 10.1186/1752-0509-6-133

Thomas, R. (1973). Boolean formalization of genetic control circuits. *J. Theor. Biol.* 42, 563–585. doi: 10.1016/0022-5193(73)90247-6

Varela, P., Mendes, N., Monteiro, P., Faure, A., and Chaouiya, C. (2013). "EpiLog: a novel tool for the qualitative modelling of epithelial patterning," in *INForum* (Évora).

Veliz-Cuba, A. (2011). Reduction of Boolean network models. *J. Theor. Biol.* 289, 167–172. doi: 10.1016/j.jtbi.2011.08.042

Zañudo, J. G. T., and Albert, R. (2013). An effective network reduction approach to find the dynamical repertoire of discrete dynamic networks. *Chaos* 23:025111. doi: 10.1063/1.4809777

Conflict of Interest Statement: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

ORIGINAL RESEARCH
published: 27 November 2018
doi: 10.3389/fphys.2018.01659

Gene Regulatory Network Modeling of Macrophage Differentiation Corroborates the Continuum Hypothesis of Polarization States

*Alessandro Palma[1], Abdul Salam Jarrah[2], Paolo Tieri[3,4], Gianni Cesareni[1,5] and Filippo Castiglione[3]**

[1] Department of Biology, University of Rome Tor Vergata, Rome, Italy, [2] Department of Mathematics and Statistics, American University of Sharjah, Sharjah, United Arab Emirates, [3] Institute for Applied Computing, National Research Council of Italy, Rome, Italy, [4] Data Science Program, Sapienza University of Rome, Rome, Italy, [5] Fondazione Santa Lucia Istituto di Ricovero e Cura a Carattere Scientifico (IRCCS), Rome, Italy

OPEN ACCESS

Edited by:
Matteo Barberis, University of Amsterdam, Netherlands

Reviewed by:
Carlos Villarreal, Universidad Nacional Autónoma de México, Mexico
Nathan Weinstein, Universidad Autónoma de Mexico, Mexico

***Correspondence:**
Filippo Castiglione
f.castiglione@iac.cnr.it

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: *31 July 2018*
Accepted: *02 November 2018*
Published: *27 November 2018*

Citation:
Palma A, Jarrah AS, Tieri P, Cesareni G and Castiglione F (2018) Gene Regulatory Network Modeling of Macrophage Differentiation Corroborates the Continuum Hypothesis of Polarization States. Front. Physiol. 9:1659. doi: 10.3389/fphys.2018.01659

Macrophages derived from monocyte precursors undergo specific polarization processes which are influenced by the local tissue environment: classically activated (M1) macrophages, with a pro-inflammatory activity and a role of effector cells in Th1 cellular immune responses, and alternatively activated (M2) macrophages, with anti-inflammatory functions and involved in immunosuppression and tissue repair. At least three different subsets of M2 macrophages, namely, M2a, M2b, and M2c, are characterized in the literature based on their eliciting signals. The activation and polarization of macrophages is achieved through many, often intertwined, signaling pathways. To describe the logical relationships among the genes involved in macrophage polarization, we used a computational modeling methodology, namely, logical (Boolean) modeling of gene regulation. We integrated experimental data and knowledge available in the literature to construct a logical network model for the gene regulation driving macrophage polarization to the M1, M2a, M2b, and M2c phenotypes. Using the software GINsim and BoolNet, we analyzed the network dynamics under different conditions and perturbations to understand how they affect cell polarization. Dynamic simulations of the network model, enacting the most relevant biological conditions, showed coherence with the observed behavior of *in vivo* macrophages. The model could correctly reproduce the polarization toward the four main phenotypes as well as to several hybrid phenotypes, which are known to be experimentally associated to physiological and pathological conditions. We surmise that shifts among different phenotypes in the model mimic the hypothetical continuum of macrophage polarization, with M1 and M2 being the extremes of an uninterrupted sequence of

states. Furthermore, model simulations suggest that anti-inflammatory macrophages are resilient to shift back to the pro-inflammatory phenotype.

Keywords: macrophage, differentiation, phenotype, model, gene regulating network, polarization, immune system

AUTHOR SUMMARY

Macrophages are key players in the elicitation of an efficient immune response. Latest classification of macrophage functional types comprises the classically activated (M1) macrophages with a pro-inflammatory activity and the alternatively activated (M2) macrophages, with anti-inflammatory functions. The latter is further subdivided into at least three different subsets, namely, M2a, M2b, and M2c, which are characterized on the basis of distinct eliciting signals.

Accounting for the gene-related mechanisms of macrophage differentiation is a challenging task. We have used the methodology known as gene regulation network modeling on a newly constructed network of gene regulation originated from published experimental data. We have used computer simulations to explore the dynamical behavior of this network and derived conclusions about the hypothetical continuum of macrophage polarization with M1 and M2 being the extremes of an uninterrupted sequences of states. Our simulations also suggest that anti-inflammatory macrophages are resilient to shift to the pro-inflammatory phenotype.

INTRODUCTION

Macrophages and neutrophils of the innate immune system represent the first line of defense against most common microorganisms. Indeed, macrophages can recognize and respond to a wide range of stimuli, expressing a great variety of surface and intracellular receptors that activate several signal transduction pathways and complex gene expression patterns. Macrophages respond to extracellular stimuli upon contact with different cell types *via* endocytic, phagocytic, and secretory functions (**Figure 1**). Their activity is modulated by contact synapsis established with proximal cellular and molecular entities, including microorganisms, chemical mediators, and other macrophages (Gordon et al., 2014).

The monocyte–macrophage differentiation pathway is known to exhibit plasticity and diversity (Mantovani et al., 2002; Bowdish et al., 2007; Gordon, 2008; Mantovani, 2008). Similar to the polarization process of helper T type 1 and 2 cells (Th1–Th2), two distinct polarized forms of macrophages have been recognized in the past: the classically activated (M1) macrophage phenotype and the alternatively activated (M2) macrophage phenotype (Biswas and Mantovani, 2010). Moreover, macrophages have also been observed in "M2-like" states, which share some features of both M1 and M2. Indeed, recent studies support the view that fully polarized macrophages (M1 and M2) are the extremes of a continuum of macrophage polarization (Mantovani, 2008). For example, various stimuli, such as immune complexes (IC) together with LPS or interleukin-1 beta (IL-1β), glucocorticoids, transforming growth factor-β (TGF-β), and interleukin-10 (IL-10), give rise to M2-like functional phenotypes that share properties with IL-4- or IL-13-activated macrophages [such as high expression of mannose receptor (MR) and IL-10, as well as TNFα, IL-1β, and IL-6] (Mantovani et al., 2004). Variations of the gene expression patterns corresponding to M1 or M2 are also found *in vivo* (e.g., in the placenta and embryo, and during helminthic infection, *Listeria* infection, obesity, and cancer) (Raes et al., 2005; Biswas et al., 2006; Kraakman et al., 2014).

The M1 and M2 phenotypes Kraakman et al., 2014 correspond to cell activation states driven by cytokines, which are typically secreted by Th1, Th2, and T-regulatory cells, but also basophils, mast cells, B lymphocytes, and eosinophils. The M1 phenotype is polarized by single or a combination of Th1 cytokines and pro-inflammatory mediators, including granulocyte-macrophage colony-stimulating factor (GM-CSF), tumor necrosis factor (TNF)-α, IL-6, IL-1β, IL-12, and various pathogen-associated molecules, such as lipopolysaccharide (LPS). By contrast, the M2 polarization is induced by macrophage colony-stimulating factor (M-CSF), IL-4 and IL-13, IC, IL-10, as well as glucocorticoid, TGFβ, and serotonin (Sang et al., 2015) (see **Table 1**).

Although there is a wealth of information about the different macrophage subsets *in vitro*, features such as plasticity, heterogeneity, and adaptability make them very difficult to study using conventional experimental tools. Furthermore, as many of the studies are done in different settings or for different goals, some literature reports are not conclusive and sometimes contradictory. It is not clear how robust the different macrophage subsets are to environmental changes. In particular, how does a modification of the cytokine environment affect the phenotype of macrophages? Which polarization state is most stable? Which possible gene knockouts can lead to a phenotypic change?

Macrophages polarization is essential in orchestrating the immune system response both in infectious and sterile immune settings. To shed light on this complex molecular process and address the questions above, we employed computational modeling of *gene regulatory networks* (GRNs) (Karlebach and Shamir, 2008).

Computational and mathematical modeling provide a means to assemble the known relevant molecules and their interactions into a network of pathways, with cross-talk between them. This allows, for examples, the test of whether the assimilated knowledge is sufficient to reproduce experimental results, and, furthermore, introduce cell-specific perturbations into the network to generate and test hypotheses *in silico*. For recent reviews, see Eftimie et al. (2016), Chakraborty (2017).

Computational models of GRNs have been shown to be a good approach to study how cells integrate several signals driving the cell phenotypic changes, especially for their ability to quantitatively and qualitatively describe a great variety of poorly

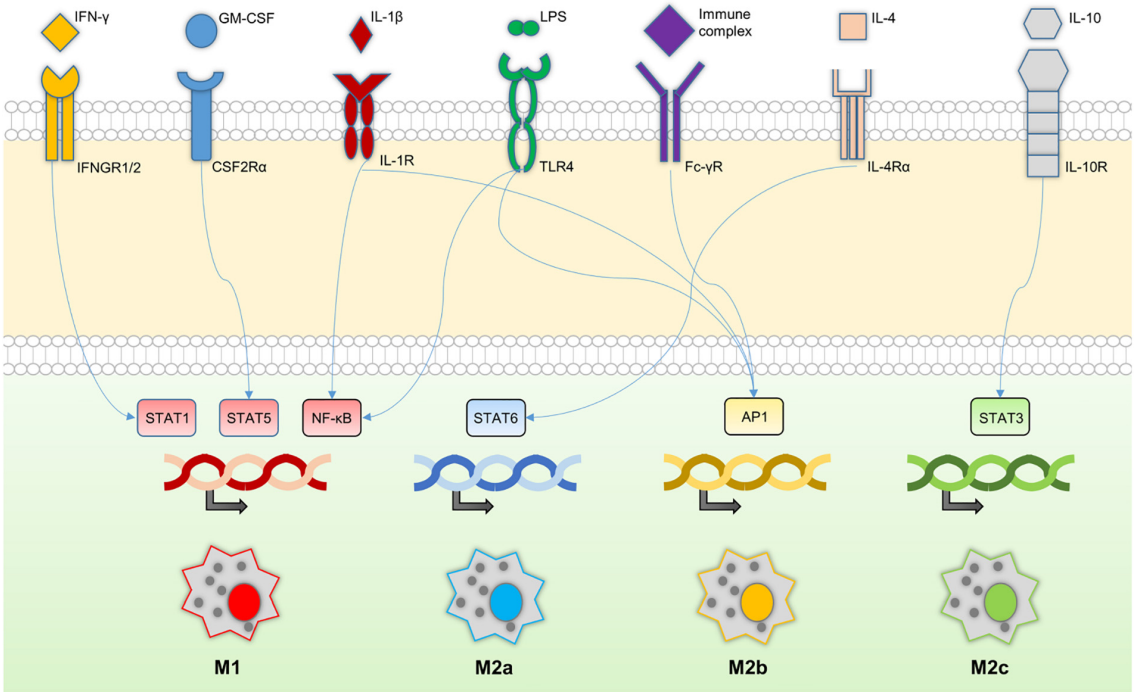


FIGURE 1 | Macrophage signaling cascade. Macrophage receptors and their relationships with key transcription factors downstream of the signaling cascade. The transcriptions of different sets of genes lead to distinctive macrophage phenotypes; M1, M2a, M2b, and M2c.

TABLE 1 | Summary of key molecules in macrophage polarization as taken from the literature.

	M1	M2A	M2B	M2C
Cytokines	IL-10, IL-1, IL-23, IL-1β, TNFα, IL-6, IL-18	IL-10, IL-12, IL-23, IL-1Ra	IL-10, IL-12, IL-23, IL-1β, TNFα, IL-6	IL-10, IL-12, IL-23, TGFβ
CC-chemokines	CCL-2, 3, 4, 5, 11, 17, 22	CCL-17, 18, 22, 24	CCL-1	CCL-16, 18
CXC-chemokines	CXCL-1, 2, 3, 5, 8, 9, 10, 11, 16	–	–	CXCL-13
Scavenger receptors	–	SR, MR	–	MR, CD163
Metabolism	iNOS	FIZZ-1, Ym-1, Arg	iNOS	Arg

Cytokines, chemokines, receptors, and genes involved in metabolism are represented for each specific macrophage phenotype (adapted from Foey, 2014). Dashes indicate missing/contrasting data in the literature.

characterized biological situations (Méndez and Mendoza, 2016). Computational models are used to describe immunological phenomena, to provide a better understanding of aspects of the immune response, and to produce outcomes coherent with available data, thus unraveling basic mechanisms of immunology and possibly leading to new hypotheses that can be tested experimentally *in vivo* or *in vitro* (Castiglione and Celada, 2015).

Discrete logical (Boolean or multi-state) models are usually the method of choice especially when the biological questions are of qualitative nature or when the available data (and knowledge) are mainly qualitative. Boolean networks and logical models have been used extensively to model many biological systems including immunological systems such as T-cell signaling and T helper cell differentiation (Naldi et al., 2010; Abou-Jaoudé et al., 2016; Méndez and Mendoza, 2016).

There are several computational models of some pathways that are involved in the pro and anti-inflammatory immune response, such as the NF-κB, TNF-α, IL-1, and IL-10 signaling pathways. Furthermore, there are computational models of T helper cell differentiation including continuous (Carbo et al., 2014), Boolean (Martinez-Sanchez et al., 2015), multistate logical (Naldi et al., 2010), and multi-scale (Santoni et al., 2008; Tieri et al., 2014). However, we are not aware of any GRN models of the molecular network describing macrophage differentiation. We have recently developed a multiscale model (Castiglione et al., 2016) of the immune response incorporating a minimalistic Boolean model of macrophages differentiation accounting for M1 and M2 polarization, but not for the subsets of M2. Maiti et al. (2014) presented an ODE model to describe the pro- and anti-inflammatory signaling in macrophages toward understanding immune homeostasis.

In this paper, we present a novel logical model of the gene regulation underlying macrophage differentiation and polarization, where the regulatory interactions and logical rules are inferred from the literature. We then used the model to study the dynamical behavior of the network. The model not only was

able to reproduce known experimental data but also provides the first computational evidence of the continuum hypothesis of phenotypes which was suggested by Sica and Mantovani (2012).

MATERIALS AND METHODS

Logical Models of Regulatory Networks

Gene regulatory network modeling aims at describing the way cells integrate extracellular stimuli to run cellular programs consisting of activations and inhibitions of genes (Kestler et al., 2008).

Logical network modeling was introduced by the geneticist R. Thomas (Thomas and D'Ari, 1990; Thomas and Kaufman, 2001) for the study of GRNs. Since then, they have been developed further, and have been used extensively to model many biological systems including cell-fate determination in *A. thaliana* (Espinosa-Soto et al., 2004; Benítez and Hejátko, 2013), *E. coli* metabolism (Samal and Jain, 2008), and the differentiation and plasticity of T helper cells (Naldi et al., 2010; Abou-Jaoudé et al., 2014), to name a few.

Gene regulatory networks are typically drawn from a mixture of literature, data mining and experimental data. Signal transducers, transcription factors and target genes in the activation of specific cellular programs (e.g., cell maturation or differentiation) are identified, as well as their relationships coded in terms of inhibition/activation. This data mining step produces a network (*N, E*) in which the nodes *N* are the molecules and the edges $E = E^- \cup E^+$ are the activations (edges in the set E^+) and inhibitions (edges in E^-) relationships. Gene activation levels (states) or molecular concentrations are represented either by a discrete and usually very small set of values (two levels, i.e., active/inactive, represents the most used one, called Boolean) or by a continuous range of activity levels. In this paper, we have used the discrete Boolean formulation.

Each node *nk* of the network *N* has a function F_k specifying how the state of that node may change in response to changes in the states of its neighbors (the nodes *nj* for which there exists an edge $ejk \in E$) in the network. The synchronous or asynchronous calculation of the functions $F_1,\ldots,F_n$, at each discrete step makes the network evolve from one macro-state to another. In the synchronous mode, all node states are updated at the same time, while in an asynchronous case, nodes are randomly updated at different time steps.

The Boolean model of a GRN is therefore defined as a discrete dynamical system which can then be studied for its dynamical properties. Since the space of all possible macro-states is finite, starting from any configuration, the repeated application of the functions $F_1,\ldots,F_n$, will lead the system to be in states that it has reached before. These states correspond to stable patterns of gene expression that can be reasonably regarded as real biological states characterizing a specific cellular function. Starting from any configuration and after a certain transient period, the network dynamics will either reach a state and stay there (such a state is called a *steady state*), or can keep cycling forever among the same set of states (such a set of states is called a *limit cycle*) (Guevara, 2003; Ortiz-Gutiérrez et al., 2015). The transient period before the network dynamics reaches a certain steady state or limit cycle is called the basin of attraction of that state or cycle.

The dynamics of the system is encoded by a graph, whose vertices are all configurations (states) of the network and directed edges where each such edge indicates the transition of the system from one state to the next.

We used the software GINsim (Naldi et al., 2009) for the development of the model and the analysis of the network, including the identification of all steady states (Karlebach and Shamir, 2008; Méndez and Mendoza, 2016), and the BooleanNet Python library (Albert et al., 2008) and BoolNet R library (Müssel et al., 2010) for the study of the dynamics of the system.

RESULTS

Molecular Basis of the Macrophage Polarization

During the inflammation process, several immune cells are involved in initiating and maintaining the inflammatory state. Macrophages, together with leukocytes, are the first cells recruited to the inflammation site. They start releasing pro-inflammatory cytokines (mostly IFN-γ and IL-1β), creating an inflammatory environment. The binding of those molecules to their specific receptors triggers a signal transduction cascade resulting in the release of other inflammatory molecules. This positive feedback mechanism allows the maintenance of the inflammatory state and reinforce the M1 polarized state.

The resolution of inflammation occurs by different mechanisms, such as the downregulation of pro-inflammatory molecules, the short half-life of the inflammatory mediators, and the production of anti-inflammatory molecules. In this context, macrophages are expected to switch to M2, and, consequently, produce anti-inflammatory mediators, such as IL-10, inhibiting M1-related transcriptional regulators, while a positive feedback loop provides the means to maintain their anti-inflammatory phenotype.

Interferon (IFN) receptors have multi-chain structures and interact with members of the Janus-activated kinase (JAK) family (Darnell et al., 1994). When IFN-γ binds to its cognate receptor, the activation of the receptor-associated JAKs occurs in response to rearrangement and dimerization of the receptor subunits, followed by auto-phosphorylation and activation of the associated JAKs. This process determines the activation of classical JAK–STAT (signal transducer and activator of transcription) signaling pathways, resulting in the transcription of target genes (Platanias, 2005; Mosser and Edwards, 2008; McLaren and Ramji, 2009). Among the STATs, a pivotal role is played by STAT1, which undergoes dimerization after its JAK-mediated tyrosine phosphorylation. Hence, STAT1–STAT1 homodimer binds to *cis* elements known as "gamma-activated sequences" (GAS) in the promoters of the genes encoding NOS2, the MHC class II transactivator (CIITA) and IL-12, among others (Darnell et al., 1994; Sadler and Williams, 2008; Lawrence and Natoli, 2011). The IFN-associated JAK/STAT pathway exerts its function in the regulation of several immune cells, including macrophages, with a great increase of IFN production, the

synthesis of several cytokines, such as interleukins IL-1β, IL-6, IL-12, IL-18, IL-23, and TNF-α, and nitric oxide (NO), as well as reactive oxide intermediates (ROI) and enzymes required for tissue remodeling.

Toll-like receptors (TLRs) mediate the immune response to a great variety of infectious agents and facilitate transcription of many pro-inflammatory genes (Sheikh et al., 2014). LPS is a component of the Gram-negative bacteria cell wall and induces expression of a wide variety of genes that constitute the innate immune response to bacterial infections. LPS signals through TLR4 on the cell surface of many cell types, including macrophages (Kawai and Akira, 2010, 2011). Signaling through TLR4 induces rapid activation of two distinct intracellular signaling pathways: one is the MyD88-dependent pathway, which leads the cascade through interferon regulatory factor (IRF)-3, and the other is the MyD88-independent signaling pathway, which acts through TIR-domain-containing adapter-inducing interferon β (TRIF). These pathways converge to activate the transcription of NOS2; the inducible NO synthase (Kawai et al., 2001; Doyle et al., 2002).

The M1 phenotype can also result from differentiation in the presence of GM-CSF, with increased expression of IL-12 and pro-inflammatory cytokines, the ability to activate Th1 cell immune responses and decreased expression of IL-10 (Krausgruber et al., 2011).

M2 macrophages exhibit a functionally distinct phenotype to that of M1s, originally *via* the ability of IL-4 to induce MR expression, followed by IL-13, which is another Th2 cytokine. IL-4/IL-13 and TGFβ/IL-10 have been described to be associated with priming M2 macrophage subsets (M2a and M2c, respectively). The role of IL-4- and IL-13-mediated signaling in M2 macrophage polarization has been well established both *in vitro* and *in vivo* (Gordon, 2003; Martinez et al., 2009; Gordon and Martinez, 2010). Mice with a myeloid cell-specific knockout of IL-4 receptor-α (IL4Rα) were found to lack M2 macrophage development in mouse models of helminth infection and in Th2 cell-mediated inflammation, where IL-4 has a major role (Lawrence and Natoli, 2011). It is well established that IL-4 and IL-13 are associated with Th2-type responses, which have well-defined effects on macrophages, other cells and immune functions. IL-4 and IL-13 are produced particularly in allergic, cellular, and humoral responses to parasitic and extracellular pathogens. IL-4 and IL-13 upregulate expression of the MR and MHC class II molecules by macrophages, which stimulates endocytosis and antigen presentation, and they induce the expression of selective chemokines (Gordon, 2003; Gordon and Martinez, 2010). IL-4 and IL-13 act through a common receptor chain – IL-4Rα – through signal transducer and activator of transcription 6 (STAT6).

Interleukin-1 beta and IC, together with TLR4-signaling inducers (i.e., LPS), drive the macrophage to an M2b phenotype. IL-1β not only plays a pivotal role in the initiation and maintenance of the inflammatory response but also modulates immunosuppressive mechanisms through the process of macrophages endotoxin tolerance. IL-1β is also produced in response to LPS, emphasizing a collaborative interplay between M1 and M2b macrophages in eliciting and maintaining the inflammatory response (Sato et al., 2012).

Interleukin-10 acts on a distinct plasma membrane receptor to those for IL-4 and IL-13 (Riley et al., 1999; Moore et al., 2001; Deng et al., 2012), and its effects on macrophage gene expression are different, involving a more profound inhibition of a range of antigen-presenting and effector functions, together with the activation of selected genes or functions. T cells themselves are more heterogeneous than was thought originally, including not only Th0-, Th1-, and Th2-type cells but also regulatory and possibly Th3-type cells, some of which secrete TGF-β and IL-10 (Gordon, 2003). TGFβ and IL-10 have been described to be associated with priming M2-like macrophages subset polarization. TGFβ and IL-10 modulate macrophage polarization and functional plasticity to that of an M2c subset which exhibits a characteristic cytokine phenotype of IL-10^{hi}, IL-12^{lo}, IL-23^{lo}, and TGFβ^{+} which is associated with anti-inflammatory responses, scavenging, immune regulation, tissue repair, and tumor promotion. Both TGFβ and IL-10 directly suppress immune activation *via* the down-regulation of the expression of MHC II and pro-inflammatory cytokine production, with an indirect effect through cross-regulation of M1-derived cytokines and functionality (Gordon and Martinez, 2010; Lawrence and Natoli, 2011; Sica and Mantovani, 2012). IL-10 is a potent STAT3-dependent inhibitor of pro-inflammatory cytokine production and NO release, after challenge with LPS. IL-10-deficient mice develop widespread inflammatory cell infiltrates, including in the bowel, and transgenic animals that constitutively overexpress IL-metricconverterProductID10 in10 in macrophages suffer from septic shock and over-activity of pro-inflammatory cytokines (Lang et al., 2002b). The upregulation of expression of IL-4Rα by IL-10 correlates with increased IL-4-dependent expression of arginase-1. IL-10 also synergizes with LPS to increase the expression of arginase-2. Therefore, IL-10 increases the total level of arginases in macrophages in many ways (Lang et al., 2002a,b).

Phenotypes depending on complex regulatory logic can be effectively studied by using mathematical and computational approaches, such as GRN models.

A Logical Network Model of Macrophage Differentiation

We have constructed a logical regulatory network model (**Figure 2** and **Supplementary File S1**) that describes macrophage polarization using experimental data and knowledge derived from literature (see **Table 2**) and a curated database of causal relationships between biological entities (Perfetto et al., 2016). The network comprises 30 nodes and 49 interactions among them. Interactions can be either positive (activations) or negative (inhibitions) (**Figure 2**). **Table 2** shows a list of the molecules, interactions, and references from the literature supporting each interaction, while **Table 3** shows logical rules for each molecule.

Nodes are of four kinds, depending on cellular location and function (**Figure 2**): seven *input* nodes, which represent the extracellular stimuli (IFNγ, GM-CSF, IL-1β, LPS, IC, IL-4, and IL-10), seven *receptors* (IFNγR, CSF2Ra, IL-1R, TLR4, FcγR, IL-4R, and IL-10R), 14 internal regulators (STAT1, STAT5, NF-κB,

TABLE 2 | Interactions in the macrophage polarization network.

Source	Interaction type	Target	Reference	Source	Interaction type	Target	Reference
IFNg_e	Positive	IFNgR	Kotenko et al., 1995; Mosser and Edwards, 2008; McLaren and Ramji, 2009	NF-κB	Positive	IL12_out	Tran-Thi et al., 1995; Lehtonen et al., 2002; Park et al., 2009; Lawrence and Natoli, 2011; Bally et al., 2015
IL1b_e	Positive	IL1R	Weber et al., 2010	NF-κB	Positive	IL1b	Tran-Thi et al., 1995; Lehtonen et al., 2002; Park et al., 2009; Lawrence and Natoli, 2011; Bally et al., 2015
GM-CSF_e	Positive	CSF2Ra	Lehtonen et al., 2002; Hamilton, 2008; Krausgruber et al., 2011; Lawrence and Natoli, 2011	PPARg	Positive	IL10_out	Ricote et al., 1998; Bouhlel et al., 2007; Lawrence and Natoli, 2011
LPS_e	Positive	TLR4	Park et al., 2009; Lawrence and Natoli, 2011				
LPS_e	Positive	FcgR	Nimmerjahn and Ravetch, 2008; Foey, 2014	PPARg	Negative	NF-κB	Ricote et al., 1998; Bouhlel et al., 2007; Lawrence and Natoli, 2011
IC_e	Positive	FcgR	Sánchez-Mejorada and Rosales, 1998; Nimmerjahn and Ravetch, 2008; Foey, 2014	PPARg	Negative	STAT3	Ricote et al., 1998; Bouhlel et al., 2007; Lawrence and Natoli, 2011
IL1b_e	Positive	FcgR	Nimmerjahn and Ravetch, 2008; Foey, 2014	STAT6	Positive	KLF4	Sica and Mantovani, 2012
IL4_e	Positive	IL4Ra	Gordon, 2003; Gordon and Martinez, 2010; Lawrence and Natoli, 2011				
IL10_e	Positive	IL10R	Moore et al., 2001; Foey, 2014; Hutchins et al., 2013; Nakamura et al., 2015	STAT6	Positive	SOCS1	Baker et al., 2009; Dickensheets et al., 2007; Whyte et al., 2011
IFNgR	Positive	STAT1	Mosser and Edwards, 2008; McLaren and Ramji, 2009	STAT6	Positive	IL10_out	Lang et al., 2002a; Gordon, 2003; Gordon and Martinez, 2010; Lawrence and Natoli, 2011
CSF2Ra	Positive	STAT5	Barahmand-Pour et al., 1998; Lehtonen et al., 2002; Hamilton, 2008; Krausgruber et al., 2011; Lawrence and Natoli, 2011	JMJD3	Positive	IRF4	Gordon, 2003; Ishii et al., 2009; Gordon and Martinez, 2010; Satoh et al., 2010; Lawrence and Natoli, 2011
IL1R	Positive	NF-κB	Weber et al., 2010	STAT3	Positive	IL10_out	Riley et al., 1999; Ritter et al., 1999; Hutchins et al., 2013; Foey, 2014; Nakamura et al., 2015
TLR4	Positive	IRF3	Sheikh et al., 2014				
TLR4	Positive	NF-κB	Tran-Thi et al., 1995; Lehtonen et al., 2002; Park et al., 2009; Lawrence and Natoli, 2011; Bally et al., 2015	STAT3	Negative	NF-κB	Riley et al., 1999; Hutchins et al., 2013
FcgR	Positive	ERK	Sánchez-Mejorada and Rosales, 1998; Sutterwala et al., 1998; Lucas et al., 2005; Nimmerjahn and Ravetch, 2008; Zhang et al., 2009; Luo et al., 2010; Clatworthy et al., 2014; Foey, 2014; Vogelpoel et al., 2014, 2015	STAT3	Negative	STAT1	Ito et al., 1999
FcgR	Negative	NF-κB	Sánchez-Mejorada and Rosales, 1998; Sutterwala et al., 1998; Ji et al., 2003; Lucas et al., 2005; Hirano et al., 2007; Nimmerjahn and Ravetch, 2008; Zhang et al., 2009; Luo et al., 2010; Clatworthy et al., 2014; Guilliams et al., 2014; Vogelpoel et al., 2014	STAT3	Negative	STAT5	Yamaoka et al., 1998
FcgR	Negative	STAT3	Sánchez-Mejorada and Rosales, 1998; Sutterwala et al., 1998; Ji et al., 2003; Lucas et al., 2005; Nimmerjahn and Ravetch, 2008; Zhang et al., 2009; Luo et al., 2010; Clatworthy et al., 2014; Guilliams et al., 2014; Vogelpoel et al., 2014, 2015	IRF3	Positive	IFNb	Doyle et al., 2002; Honda et al., 2005; Rauch et al., 2013; Mao et al., 2015
FcgR	Negative	TLR4	Sánchez-Mejorada and Rosales, 1998; Sutterwala et al., 1998; Abrahams et al., 2000; Nimmerjahn and Ravetch, 2008; Zhang et al., 2009; Luo et al., 2010; Guilliams et al., 2014; Vogelpoel et al., 2014, 2015	ERK	Positive	IL10_out	Sánchez-Mejorada and Rosales, 1998; Lucas et al., 2005; Nimmerjahn and Ravetch, 2008; Liu et al., 2009; Foey, 2014

(Continued)

TABLE 2 | Continued

Source	Interaction type	Target	Reference	Source	Interaction type	Target	Reference
IL4Ra	Positive	PPARg	Gordon, 2003; Bouhlel et al., 2007; Chawla, 2010; Gordon and Martinez, 2010; Gong et al., 2012	KLF4	Negative	NF-κB	Sica and Mantovani, 2012
IL4Ra	Positive	STAT6	Gordon, 2003; Ishii et al., 2009; Gordon and Martinez, 2010; Satoh et al., 2010; Lawrence and Natoli, 2011	SOCS1	Negative	STAT1	Dickensheets et al., 2007; Baker et al., 2009; Whyte et al., 2011
IL4Ra	Positive	JMJD3	Gordon, 2003; Ishii et al., 2009; Gordon and Martinez, 2010; Satoh et al., 2010; Lawrence and Natoli, 2011	IRF4	Negative	STAT5	Sica and Mantovani, 2012
IL10R	Positive	STAT3	Riley et al., 1999; Ritter et al., 1999; Hutchins et al., 2013; Foey, 2014; Nakamura et al., 2015				
STAT1	Positive	IL12_out	Mosser and Edwards, 2008; Sadler and Williams, 2008; McLaren and Ramji, 2009; Lawrence and Natoli, 2011	IFNb	Positive	IFNgR	Kotenko et al., 1995; Lehtonen et al., 2002; Gordon, 2003; Platanias, 2005; Lawrence and Natoli, 2011; Rauch et al., 2013
STAT5	Positive	IL12_out	Yamaoka et al., 1998; Lehtonen et al., 2002; Hamilton, 2008; Krausgruber et al., 2011; Lawrence and Natoli, 2011				

Source and target nodes are reported as well as the sign of the interaction between them (positive: source molecule activates target molecule; negative: source molecule inhibits target molecule) and the references. Each input node is annotated with an "_e" suffix, which stands for external stimulus, as well as an "_out" suffix which stands for output.

PPARγ, STAT6, JMJD3, STAT3, IRF3, ERK, KLF4, SOCS1, IRF4, IL1β, and IFN-β), and two main *products* of each distinct type of macrophage (IL-12 and IL-10). The input nodes represent the main intercellular molecular stimuli that drive macrophage polarization, as reported in the literature. Each external molecule (input) is connected to its specific receptor, and this binding elicits a signaling cascade, involving intracellular transducers and transcription factors (mostly STAT factors). Each specific transcription factor binds the promoter of a target gene, resulting in the production of IL12 or IL10 depending on the macrophage polarized form.

Interactions among nodes are derived from experimental data available in the literature as shown in **Table 2**. All interactions have been deposited in SIGNOR (Perfetto et al., 2016), a public database of causal interactions between biological entities. Each node is associated to a logical function which determines the activation level of the node based on the activation levels reached by its source nodes in the previous time step. The logical function of each node is inferred from the available literature (see **Table 3**).

The network encompasses several pathways. Different cell fates, i.e., macrophage phenotypes, are defined by *steady* or *stable states* (also called *fixed point attractors*) of gene expression, and described in this dynamic model as multiple, specific, and stable configurations of activated/deactivated nodes. In other words, stable states are configurations toward which the system tends to evolve, for a wide range of starting conditions. Thus, according to the network, its starting configuration, and the initial external stimuli, the pathways lead to a configuration that resembles a specific cell state in terms of the given gene expression pattern. In this regard, we assumed that the sum of the sizes of the basins of attraction of the steady states characterizes the likelihood of finding the cell in a specific differentiation state. In other words, the probability that the cell, stimulated by cytokines, will switch to the certain differentiation state is proportional to the size of the subspace of all network configurations eventually reached by the network dynamics.

Inhibitory pathways among M1 and M2 phenotype-related transcription factors are particularly interesting, because they allow a mutual exclusivity of transcription factors and, therefore, of the macrophage phenotypes, as reported in literature (Lawrence and Natoli, 2011). Notably, among the interactions describing the network and reported in tables above, the inhibition of TLR4 and NF-κB signaling by FcγR activation were added. These relationships allow the inhibition of M1 polarization in the presence of IC, that together with LPS and IL-1β, drives the otherwise absent M2b polarization.

To analyze the dynamics of the network under different conditions we used GINsim [Gene Interaction Network simulation[1]; (Chaouiya et al., 2012)], a software tool for modeling and simulation of genetic regulatory networks (Chaouiya et al., 2012). In some cases, for further confirmation or additional details, we used the BooleanNet Python (Albert et al., 2008) as well as the BoolNet R library (Müssel et al., 2010).

The fate of a macrophage strongly depends on the local biochemical microenvironment. To reproduce these different microenvironments that influence the cells, we defined a set of inputs to run the simulations. Hence, we could discriminate among steady states with a real biological meaning.

The starting expression state of the network corresponds to the naïve macrophage M0 (unstimulated/not-activated) phenotype, in which the state of each node in the network is set to "0" (i.e., low expression).

[1]www.ginsim.org

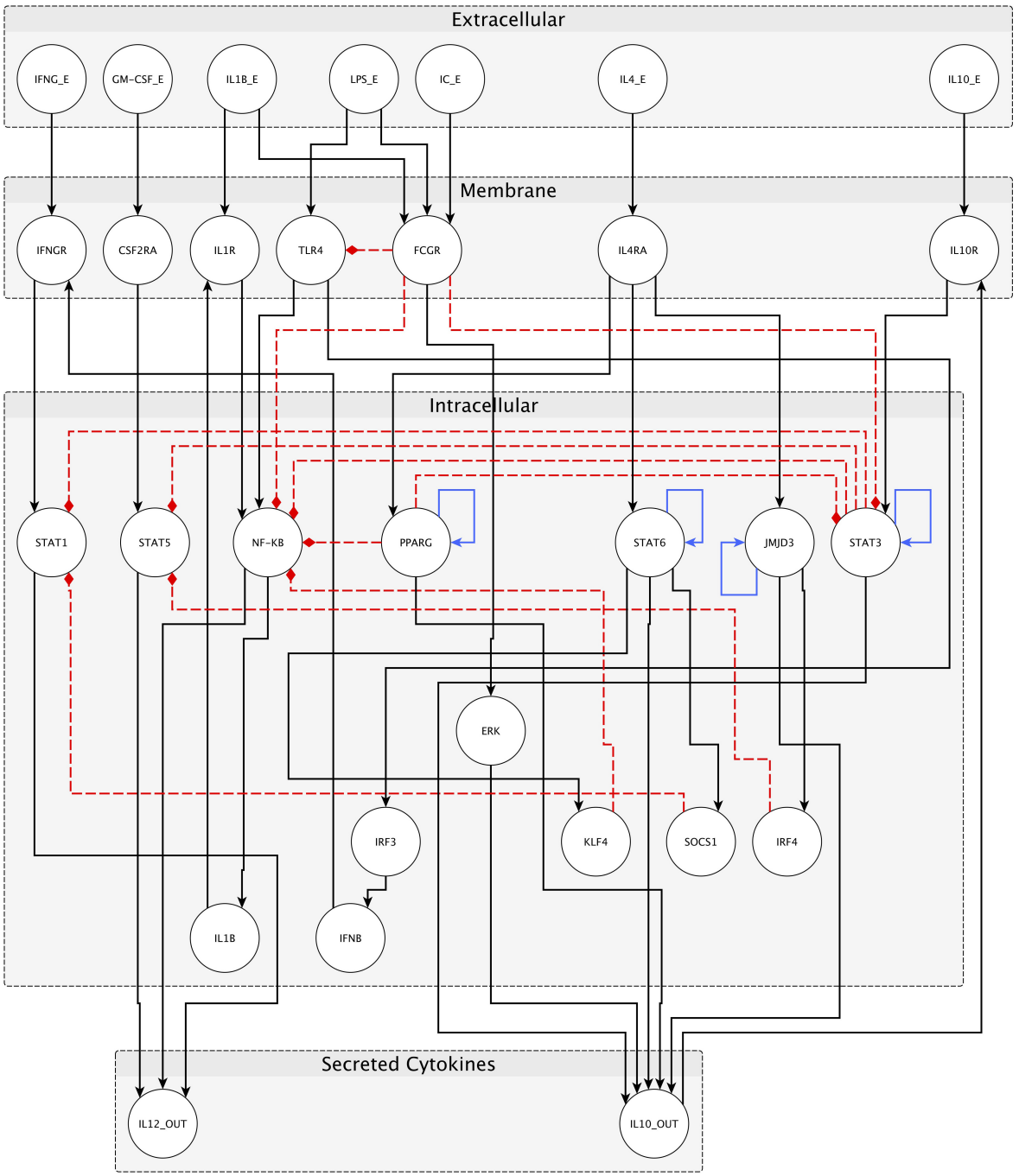


FIGURE 2 | Network for macrophage polarization. External stimuli are reported in the extracellular space, receptors inside membrane space, and internal transducers/transcription factors in the intracellular space. Secreted cytokines (IL-10 and IL12) are also reported. Black arrows represent positive interactions (activations), red dashed arrows are negative interactions (inhibitions), and blue arrows are transcriptional auto-regulatory loops. Nodes represent both genes and proteins; edges represent both protein–protein interactions and transcriptional regulations.

In our simulations, we found that our model has five sets of steady states fitting the following five specific macrophage phenotypes markers according to literature (**Figure 3**):

1. M0: no nodes active;
2. M1: IL-12 and at least one among STAT1, STAT5 or NF-κB are active;
3. M2a: all of PPARγ, STAT6, JMJD3 and IL-10 are active;
4. M2b: ERK and IL-10 are active; and
5. M2c: STAT3 and IL-10 are active.

We computed the steady states of macrophage polarization network using a synchronous update. The system reached 1056 states, 1040 of which are steady states and 16 are cycles made

TABLE 3 | Boolean functions in the macrophage polarization network.

Node	Boolean function	Reference
IFNgR	IFNg_e ∨ IFNb	Interferons bind to their cognate receptors (Kotenko et al., 1995; Lehtonen et al., 2002; Gordon, 2003; Platanias, 2005; Mosser and Edwards, 2008; McLaren and Ramji, 2009; Rauch et al., 2013)
CSF2Ra	GM-CSF_e	GM-CSF ligand binds to its receptor (Lehtonen et al., 2002; Hamilton, 2008; Krausgruber et al., 2011; Lawrence and Natoli, 2011)
IL1R	IL1b_e ∨ IL1b	IL-1 beta binds to its receptor (Weber et al., 2010)
TLR4	LPS_e ∧ ¬ FcgR	TLR4 is activated by LPS; TLR4 signaling is inhibited by Fc gamma receptor (Sánchez-Mejorada and Rosales, 1998; Sutterwala et al., 1998; Nimmerjahn and Ravetch, 2008; Park et al., 2009; Zhang et al., 2009; Luo et al., 2010; Lawrence and Natoli, 2011; Vogelpoel et al., 2014)
FcgR	(IC_e ∧ LPS_e) ∨ (IC_e ∧ IL1b_e)	Immune complexes, together with LPS or IL-1 beta activate Fc gamma receptor (Sánchez-Mejorada and Rosales, 1998; Sutterwala et al., 1998; Abrahams et al., 2000; Lucas et al., 2005; Nimmerjahn and Ravetch, 2008; Zhang et al., 2009; Luo et al., 2010; Clatworthy et al., 2014; Guilliams et al., 2014; Vogelpoel et al., 2014, 2015)
IL4Ra	IL4_e	IL-4 binds to its receptor (Gordon, 2003; Gordon and Martinez, 2010; Lawrence and Natoli, 2011)
IL10R	IL10_e ∨ IL10_out	IL-10 binds to its receptor (Moore et al., 2001; Hutchins et al., 2013; Foey, 2014; Nakamura et al., 2015)
STAT1	IFNgR ∧ ¬(SOCS1 ∨ STAT3)	Interferon-gamma receptor activates JAK/STAT1 pathway and is inhibited by SOCS1 or STAT3 signaling (Ito et al., 1999; Dickensheets et al., 2007; Mosser and Edwards, 2008; Baker et al., 2009; McLaren and Ramji, 2009; Whyte et al., 2011)
STAT5	CSF2Ra ∧ ¬(STAT3 ∨ IRF4)	STAT5 transcription factor is activated *via* CSF2Ra signaling and inhibited by STAT3 or IRF4 (Barahmand-Pour et al., 1998; Ito et al., 1999; Lehtonen et al., 2002; Dickensheets et al., 2007; Hamilton, 2008; Baker et al., 2009; Krausgruber et al., 2011; Lawrence and Natoli, 2011; Whyte et al., 2011)
NF-κB	(IL1R ∨ TLR4) ∧ ¬(STAT3 ∨ FcgR ∨ PPARg ∨ KLF4)	NF-κB transcription factor is activated by LPS or IL1-beta signaling cascades and inhibited by M2a- or M2b-related pathways (Tran-Thi et al., 1995; Ricote et al., 1998; Sánchez-Mejorada and Rosales, 1998; Sutterwala et al., 1998; Riley et al., 1999; Lehtonen et al., 2002; Bouhlel et al., 2007; Nimmerjahn and Ravetch, 2008; Park et al., 2009; Zhang et al., 2009; Luo et al., 2010; Weber et al., 2010; Lawrence and Natoli, 2011; Sica and Mantovani, 2012; Hutchins et al., 2013; Guilliams et al., 2014; Vogelpoel et al., 2014; Bally et al., 2015)
PPARg	IL4Ra	PPARg is activated by IL4 signaling (Gordon, 2003; Bouhlel et al., 2007; Chawla, 2010; Gordon and Martinez, 2010; Gong et al., 2012)
STAT6	IL4Ra	JAK/STAT6 pathway is activated by IL4 receptor after IL-4 binding (Gordon, 2003; Ishii et al., 2009; Gordon and Martinez, 2010; Satoh et al., 2010; Lawrence and Natoli, 2011)
JMJD3	IL4Ra	JMJD3 is activated in response to IL4 signaling cascade (Gordon, 2003; Ishii et al., 2009; Gordon and Martinez, 2010; Satoh et al., 2010; Lawrence and Natoli, 2011)
STAT3	IL10R∧ ¬(FcgR ∨ PPARg)	JAK/STAT3 pathway is activated in response to IL-10 and inhibited by PPAR gamma or Fc gamma receptor pathways (Ricote et al., 1998; Sánchez-Mejorada and Rosales, 1998; Sutterwala et al., 1998; Riley et al., 1999; Ji et al., 2003; Bouhlel et al., 2007; Nimmerjahn and Ravetch, 2008; Lawrence and Natoli, 2011; Hutchins et al., 2013; Foey, 2014; Nakamura et al., 2015)
IRF3	TLR4	IRF3 is activated in response to TLR4 signaling pathway (Doyle et al., 2002; Sheikh et al., 2014; Mao et al., 2015)
ERK	FcgR	ERK pathway is initiated in response to M2b-related signals (Sánchez-Mejorada and Rosales, 1998; Lucas et al., 2005; Nimmerjahn and Ravetch, 2008; Liu et al., 2009; Foey, 2014)
KLF4	STAT6	KLF4 is activated downstream JAK/STAT6 pathway (Sica and Mantovani, 2012)
SOCS1	STAT6	SOCS1 is activated by STAT6 transcription factor (Baker et al., 2009; Whyte et al., 2011; Arnold et al., 2014)
IRF4	JMJD3	IRF4 is activated by JMJD3 expression (Gordon, 2003; Ishii et al., 2009; Gordon and Martinez, 2010; Satoh et al., 2010; Lawrence and Natoli, 2011)
IL1b	NF-κB	NF-κB transcription factor promotes IL-1 beta production (Tran-Thi et al., 1995; Lehtonen et al., 2002; Park et al., 2009; Lawrence and Natoli, 2011; Bally et al., 2015)
IFNb	IRF3	IRF3 promotes type I interferon production (Doyle et al., 2002; Honda et al., 2005; Rauch et al., 2013; Mao et al., 2015)
IL12_out	STAT1 ∨ STAT5 ∨ NF-κB	IL-12 is produced by transcription factors STAT1, STAT5 or NF-κB (Mosser and Edwards, 2008; Sadler and Williams, 2008; McLaren and Ramji, 2009; Lawrence and Natoli, 2011)
IL10_out	PPARg ∨ STAT6 ∨ JMJD3 ∨ STAT3 ∨ ERK	PPAR gamma, STAT6, JMJD3, STAT3 and ERK downstream genes lead to the production of high quantities of IL10 (Ricote et al., 1998; Sutterwala et al., 1998; Riley et al., 1999; Ritter et al., 1999; Lang et al., 2002a; Gordon, 2003; Lucas et al., 2005; Bouhlel et al., 2007; Ishii et al., 2009; Liu et al., 2009; Gordon and Martinez, 2010; Luo et al., 2010; Satoh et al., 2010; Lawrence and Natoli, 2011; Foey, 2014; Sanin et al., 2015)

Based on the available literature (third column), a Boolean function (second column) is associated to each target node of the network (symbols ∧, ∨, and ¬ indicate logical operators AND, OR, NOT, respectively).

of three different states. Among the 1040 unique steady states (**Supplementary Table S1**), 228 can be mapped to the five *canonical* macrophage phenotypes reported *via* experimental studies in the literature. The frequencies of these 228 steady states are reported in **Figure 4**. The remaining steady states do not characterize the macrophage in any of the known canonical phenotypes. These states, for which there is a lack of experimental knowledge, could correspond to input conditions not existing among *in vivo* inflammation settings or even be artefacts of the modeling approach. Alternatively, they could correspond to *hybrid* phenotypes (O'Carroll et al., 2013) resembling gene expression patterns of two or more canonical phenotypes (discussed below). It is worth to note that a higher number of steady states does not imply a corresponding higher probability of polarization, since the final outcome depends on the combination of external stimuli. In other words, the number of steady states indicates the propensity of the network logic to lead the cell to the specific phenotypes yet driven by environmental cues.

The most frequent polarized state is the M2a followed by M2c and then M1. This is consistent with the pivotal role of macrophages in inflammation (M1), and in the resolution of inflammation (M2a and M2c). On the other hand, according to our analysis, M2b is the least frequent state, which might be consistent with the lack of knowledge of M2b-related pathways which is reflected in the network. This behavior of the model is consistent with observed data (Sica and Mantovani, 2012).

A closer look at the dynamics of the model (**Figure 5**) is obtained by performing several rounds of asynchronous simulations by using the BooleanNet Python library. We observed that any combination of stimuli among IFN-γ, IL-1β, LPS, and GM-CSF keep the polarization of the M1 macrophage. Once macrophages have polarized into an M1 form, the steady states are taken as initial conditions to polarize macrophages into the three different forms of M2 macrophage. IL-4 input is activated (i.e., IL-4 binding by IL-4RA) to polarize M2a macrophage, IL-10 is activated to polarize the M2c macrophages, and IC in combination with either IL-1β or LPS is activated to polarize M2b macrophages, according to the available literature on macrophage polarization stimuli (Gordon and Martinez, 2010).

The M1 polarization is simulated starting from an M0 (i.e., all non-input signal nodes set to zero) cellular environment and switching on all input nodes, as reported in literature. Following the typical cellular response to inflammation, starting from an M1-like configuration, and M2-related external stimuli (i.e., IL-4 for M2a, IL-10 for M2c and IC in combination with LPS or IL-1β for M2b macrophages), the dynamics of transcription factors and secreted molecules (i.e., IL-12 and IL-10) show the macrophage moves from pro- to anti-inflammatory states, as reported in literature. The M2-related polarizations from an M0 initial state have been also performed to check the ability of the system to simulate the situation in which new monocyte-derived macrophage populations are recruited to the inflammation site during the resolution of inflammation, in addition to M2 macrophages polarized from the pro-inflammatory M1 state (see **Supplementary File S2**).

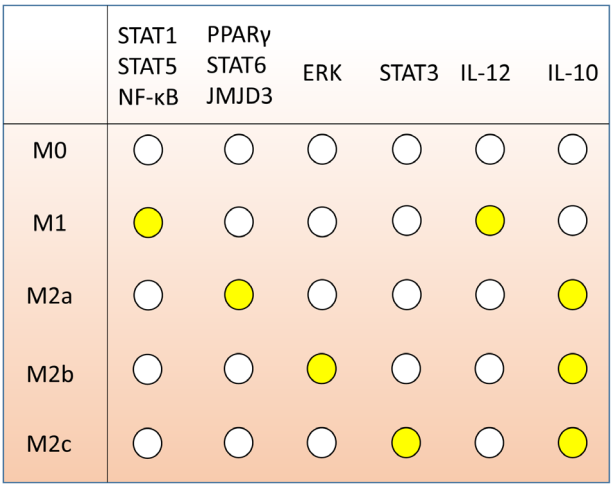


FIGURE 3 | Gene expression markers of macrophage polarization according to literature. Each row, associated to one of M0, M1, M2a, M2b, and M2c, indicates the expression of the 10 marker genes determining the polarization fate. White dots represent inactive genes; yellow dots indicate expressed genes.

We also tested *in silico* the "plasticity" of the polarized phenotypes, i.e., the capability to revert the state from inflammatory to anti-inflammatory and *vice versa*. In order to proceed, we run a set of numerical experiments in which macrophages, starting from the four polarized states M1, M2a, M2b, and M2c, were challenged with the four characteristic stimuli (i.e., pro-M1, -M2a, -M2b, and -M2c) resulting in 16 possible couples "initial condition/stimuli." Each of those simulation settings was repeated 10^4 times using the asynchronous updating scheme and averages were computed. After that, we used the steady states obtained as initial states for other simulations, giving each input from the input set (see **Figure 5**).

We focused on M1-related initial states, since a normal immune response begins with an inflammation state, followed by anti-inflammatory environment settings.

With an M0 steady state as initial condition, several stimuli were applied for each simulation. To represent the M1 polarization we gave a combination of random M1-related stimuli (LPS, GM-CSF, IFN-γ, and IL-1β). The initial state for each node of the network are those related to the M0 steady state (no active nodes at all).

We then performed M2a, M2b, and M2c polarizations with IL-4, a combination of IC and IL-1β or LPS, and IL-10 as inputs, respectively. In other words, we started with M1 macrophages, changed their environment and stimulated them with different types of stimuli. Thus, we performed all the combinations for the simulations and analyzed the dynamics and the differences (see **Figure 5** for details). We also investigated the possibility of transforming an M2-like phenotype to an M1 macrophage by

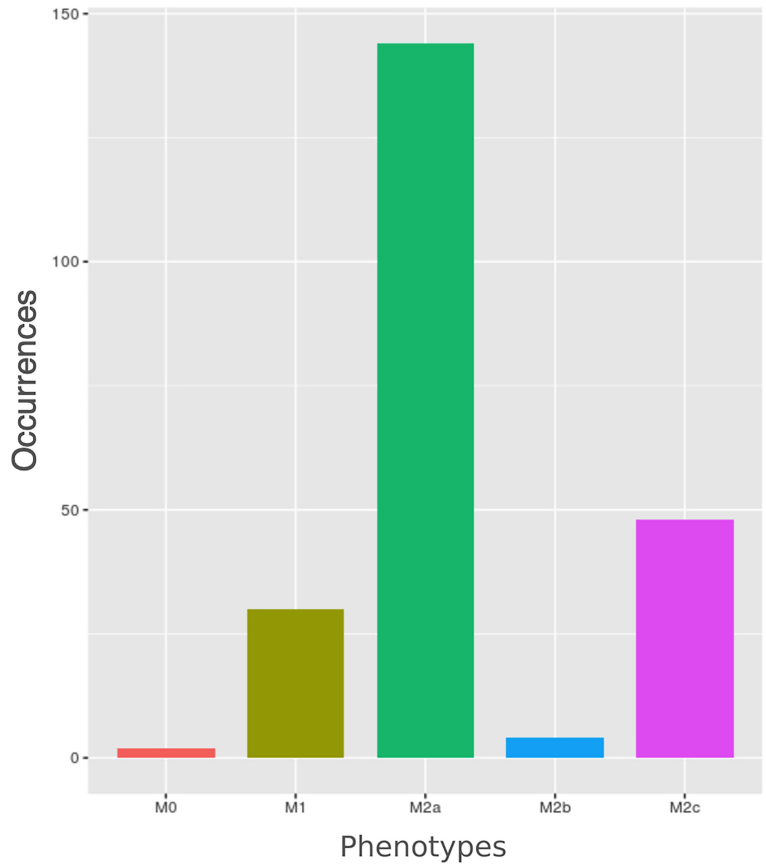


FIGURE 4 | Barplot of macrophages' phenotypes occurrences. Each bar represents the number of steady states (total number = 228) related to a specific polarized form.

changing the environment using a variety of external stimuli. However, all considered combinations resulted in states that do not characterize the macrophage M1 canonical phenotype.

Robustness Evaluation of the Macrophage Network

Biological networks are considered to be robust when compared to random networks, if a single perturbation does not influence the behavior of the entire system. We analyzed the robustness of macrophage polarization network as follows. First, we evaluated the *transition robustness* by perturbing states of the network with random bit flips (Müssel et al., 2010). When the successor states of the original and the perturbed states are computed, the distance between then is calculated as the Hamming distance (HD, that is, the difference between strings of equal length is the number of positions at which the corresponding symbols are different). The HD, normalized by the number of genes in the network, shows how robust the network is to small mutations: the lower the normalized HD, the more robust is the network.

A hundred of these tests were repeated for 100 randomly generated networks and the results plotted in **Figure 6**. Results show that the macrophage model is statistically more robust (p = 0.01) in comparison to the randomly generated networks. The resulting mean normalized HD equal to 0.03 can be interpreted as if, on average in the mutated networks, 3% of the gene states are different.

Effects of Knockouts in the Simulations

To analyze the dynamics and investigate the role of each component in the polarization process, we performed knockout (components' value set to "0") and ectopic expression (components' value set to "1") *in silico* experiments. These constraints allowed us to see how perturbations of the system affect the network functionality with respect to the macrophage behavior. At a biological level, this analysis may have potential impact in in-silico pharmaceutical target prioritization.

In our network, gene knockout is interpreted as a deactivation of one or more components, just like the deactivation of a protein that is a target of a drug.

We performed systematic knockouts on every internal node of the network (internal transducers/transcription factors), to see how they affect the dynamics of the network by calculating the fold change of the number of steady states reached by the system (see **Figure 7** and **Supplementary Files S3**, **S4** for details). The idea is that a knockout modifies the network characteristics so that also its dynamics is modified and the number of steady states,

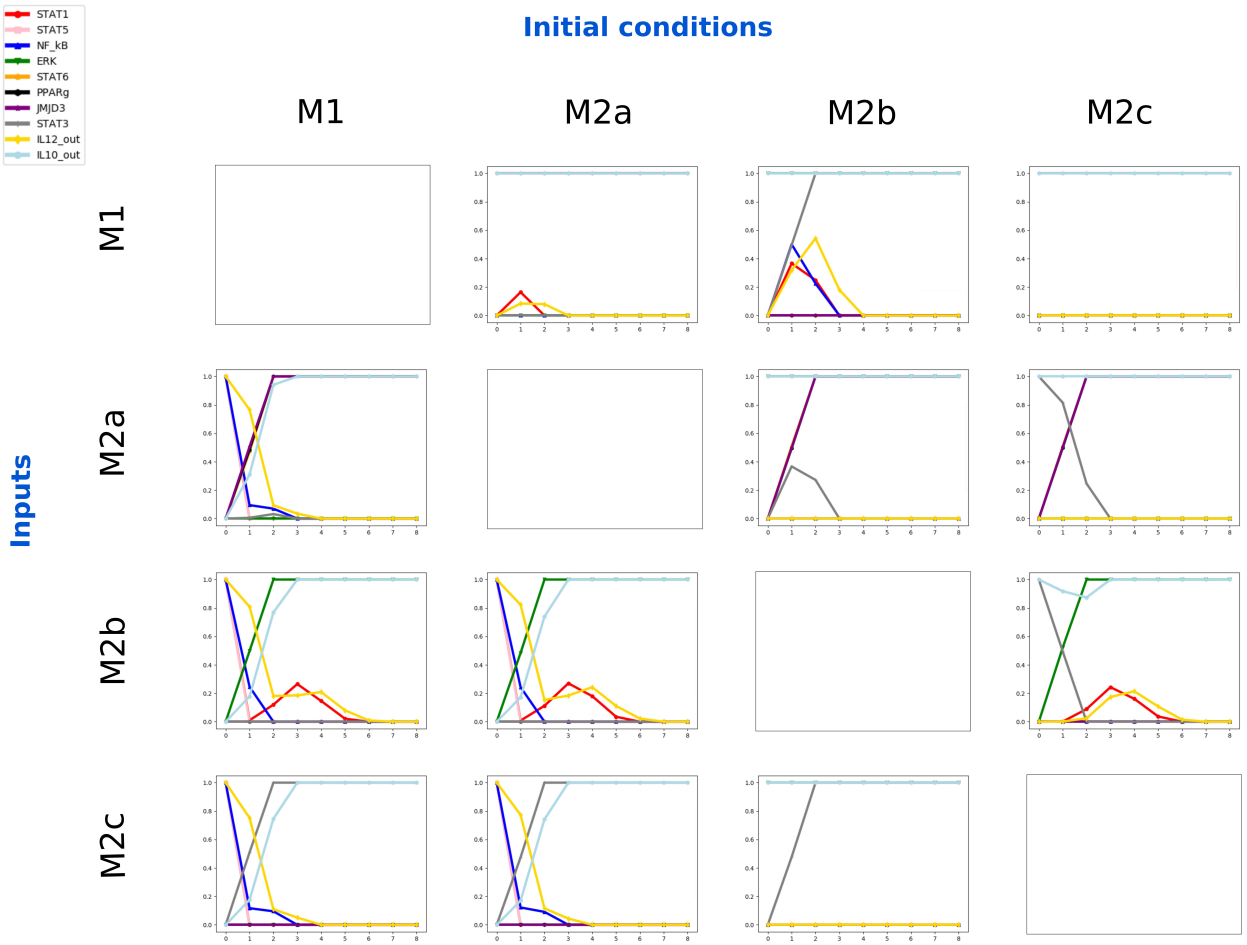


FIGURE 5 | Dynamics of the gene activation levels obtained for all combinations of initial polarization state and polarizing stimuli. The average activation values are computed over 10^4 asynchronous simulations of the activation level of the genes. For each subplot, the horizontal axis represents eight time steps and the vertical axis the average activity of a molecule from 0 to 1.

for example, a higher number of pro-inflammatory steady states is interpreted as a greater probability to induce, *via* that specific knockout, a pro-inflammatory polarization of the macrophages.

DISCUSSION

Pro-inflammatory macrophages are those polarized by cytokines like IFN-γ or LPS (among other molecules). They are produced during cell-mediated immune responses, interacting with chemical mediators produced by other cells, such as the IFN-γ secreted by natural killer (NK) cells (Mosser and Edwards, 2008). Resting macrophages are primed by IFN-γ to produce pro-inflammatory cytokines, according to our simulations of an unstimulated macrophage which undergoes an M1 polarization when stimulated by IFN-γ (see **Figure 5**). TLR ligands, such as the well-known LPS can also polarize macrophage into an M1 form, *via* NF-κB signaling, producing pro-inflammatory mediators, other stimuli such as GM-CSF and IL-1β gave similar results (Mosser and Edwards, 2008; Lawrence and Natoli, 2011; Sica and Mantovani, 2012). Macrophages respond to micro-environmental cues, showing a distinct transcriptional profile depending on the stimulus. Starting from M0, that is assumed to be a cell with no typical constitutive gene expression profile, an M1 stimulus (i.e., IFN-γ, LPS, IL-1β, and GM-CSF) leads to a M1 phenotype, IL4 to a M2a phenotype, IC together with LPS and/or IL-1β to an M2b phenotype, and IL-10 to a M2c phenotype, the network can represent the polarization process (see **Figure 8** for a visual representation of macrophage switch pathways).

Transcription factor NF-κB is among the most important regulators of M1 polarization of macrophages (Wang et al., 2014). Its expression is stable and maintained during macrophage polarization after stimulation with M1-related inputs. If no inputs are given to an M1-polarized system, NF-κB seems to maintain the M1 polarization (see **Figure 5**), while STAT1 and STAT5 decrease their expression (if not stimulated by IFN-γ and GM-CSF), until an M2-related stimulus (IL-4, IL-10 or IC) is present, which result in the resolution of the inflammation phase, and in the increase of the expression of M2 master regulators.

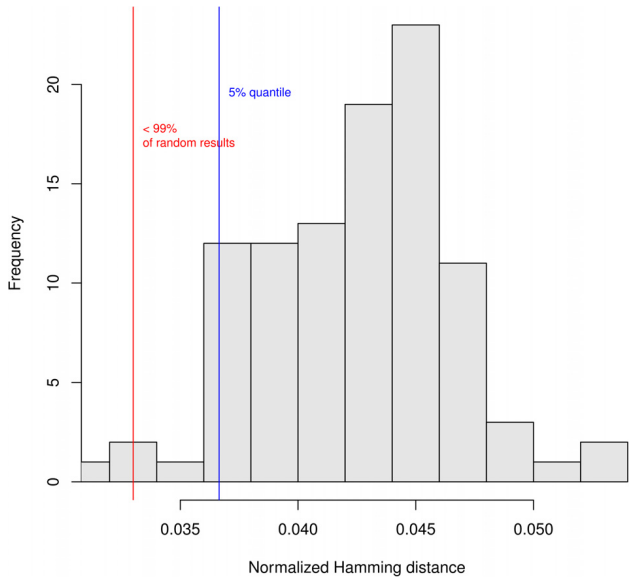


FIGURE 6 | Test of the robustness of the macrophage network. Histogram of the normalized Hamming distance (HD) of randomly generated networks (RGN) in comparison to the HD of the perturbed macrophage network (PMN). The red line shows the mean of the PMN-HD histogram (not shown) which is smaller than the 5% quantile of the RGN-HD distribution (blue line). The test shows that the noise influences the randomly generated networks significantly more than the macrophage network (Müssel et al., 2010).

In the presence of IL-4 (i.e., activation of input node IL4), we noticed rapid expression of M2a master regulators (i.e., STAT6, PPARγ, and JMJD3) and the production of IL-10, with a slow decrease in the production of IL-12, indicating that M2a-related stimuli can immediately suppress the pro-inflammatory function of macrophage, as already evidenced in literature (Sica and Mantovani, 2012). In M2b polarization, despite the slow decrease of the expression of pro-inflammatory transcription factors and secreted molecules, IL-10 is finally produced by this type of macrophage, and its master regulator, ERK. M2c polarization is reached when IL-10 is given as input, with IL-10 production and STAT3 expression.

In the absence of external stimuli, a polarized M2 macrophage maintained its state with no alteration on the molecules expression, highlighting the stability of this phenotype.

M1 stimuli do not affect M2-like macrophage, apart from M2b in which we can assist to a slower decrease of IL12, reaching its stable state at the seventh time step, at variance with M2a and M2c simulations in which the anti-inflammatory stimuli lead to the absence of IL12 at the fourth time step. For any input given to an M2b-polarized macrophage, a phenotype change related to the given stimulus seems to be a common feature, except for M1 stimuli, which appear to polarize macrophage to a form corresponding to the production of both output cytokines (IL12 and IL10) and the repression of ERK. This behavior has not been reported in literature, but could explain the existence of this not-well characterized type of macrophage that share common features between pro- and anti-inflammatory macrophages (Sica and Mantovani, 2012).

A similar behavior can be observed when M2c macrophage are polarized with M1-related cytokines, even though M2a and M2b stimulations can subvert M2c polarization, indicating that M2c macrophages are more likely to be polarized from an M0 phenotype or switch from an already M1-polarized macrophage. Indeed, in some physiological and pathological conditions, such as muscle regeneration, the co-existence of different populations of M2 macrophages can be found at later stages, comprising M2a and M2c macrophage (Novak et al., 2014; Rigamonti et al., 2014). Hence, they can be thought of as distinct populations of macrophage polarized independently, since this regulatory network is characterized by well-known interactions between molecules involved in the polarization pathway (Novak et al., 2014; Rigamonti et al., 2014).

CONCLUSION

Transforming acute diseases into chronic ones is a realistic strategy for those pathologies for which no definitive cure is known, such as in the case of HIV (Scandlyn, 2000). A better understanding of the pathways involved in the transition from acute to chronic states and a more comprehensive knowledge of the cellular and molecular mechanisms are in need. Understanding how the immune response is regulated, and how immune cells integrate information from the multitude of molecular signals could certainly lead to improvements of existing therapies and make suggestions on the way forward.

In this work, we presented a dynamic logical model of the GRN of macrophage polarization, which is coherent to the expected behavior, under different experimental conditions. The model identified mechanisms driving a pro- into an anti-inflammatory setting, and hence maybe useful in transforming, fully or in part, an acute inflammation into a chronic one.

One example of network dynamics that could be affected by providing different types of stimuli is reported in **Figure 8**. We examined the different dynamics of this process to study how macrophages switch their phenotype during ineffective and sterile immune responses, focusing on M2-like polarization from a pro-inflammatory micro-environment.

A first result regards the importance of two inhibitions, namely, of TLR4 and NF-κB signaling by FcγR, that turned out *essential* to obtain the M2b phenotype. In fact, a preliminary version of the network, not accounting for these two inhibitions, was not able to reach the M2b polarized state.

The repolarization from M2 to M1 has been experimentally observed, yet occasionally in specific environments (Davis et al., 2013; Zheng et al., 2013; Zhang et al., 2017; Gao et al., 2018). Simulation results suggest that such polarization reversion seems to show a higher inertia. In fact, as shown in **Figure 5** panels a, b, and c, the average values of pro-inflammatory genes starting from an anti-inflammatory phenotype only reach the value of 30% of the activation level. Furthermore, our *in silico* knockout experiments evidenced how some regulator plays a role by downregulating genes that are known for their inhibition activity.

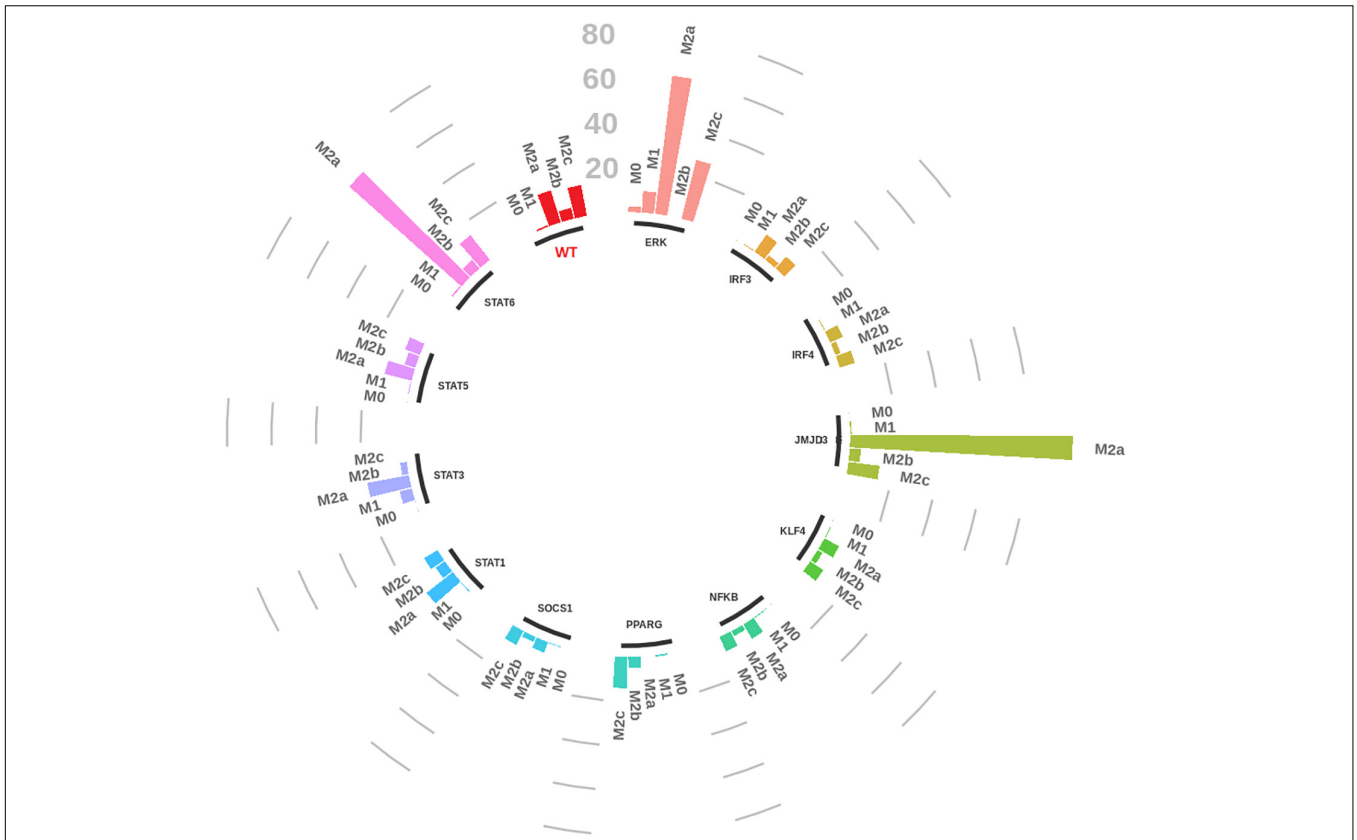


FIGURE 7 | Circular bar plot of macrophage gene knockouts. Each group represents the knockout of a specific transcription factor of the network. Bar heights represent the number of steady states for each macrophage canonical phenotype with respect to the wild type (WT in red).

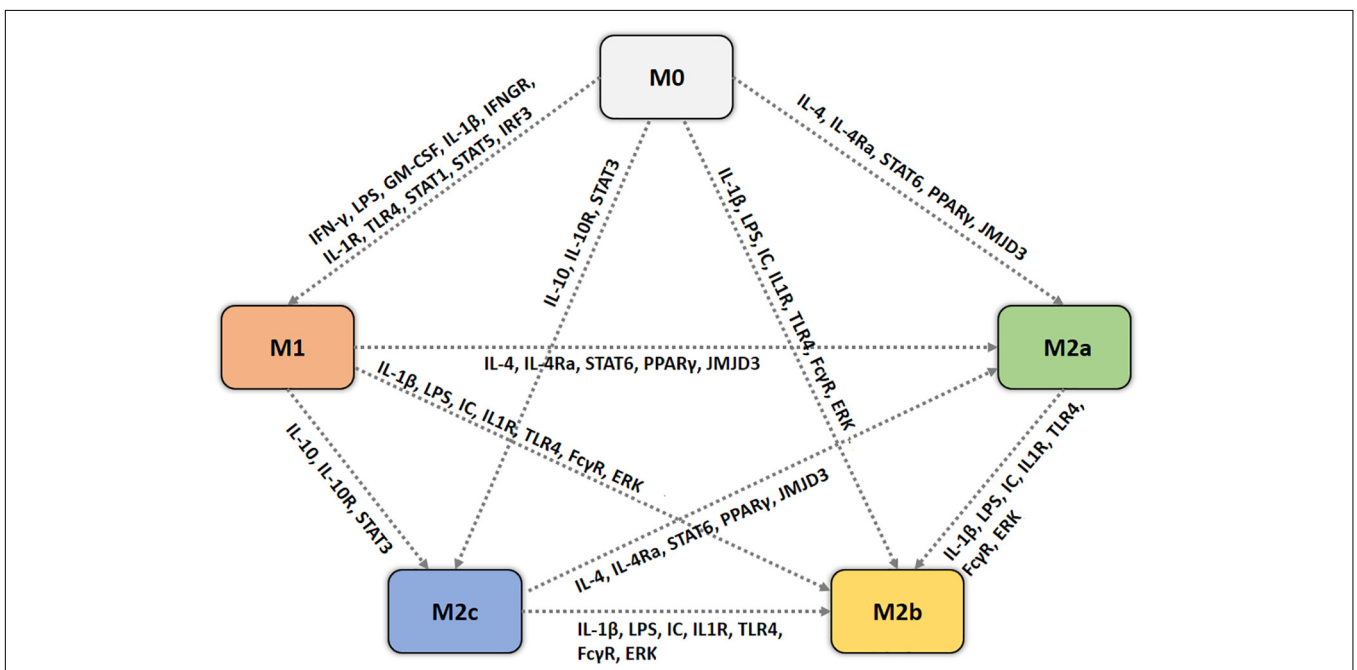


FIGURE 8 | Cell fate map for macrophages. Each dotted arrow represents the switch of macrophage from a phenotype to another, annotated with the gene expression patterns, based on simulation dynamics and results.

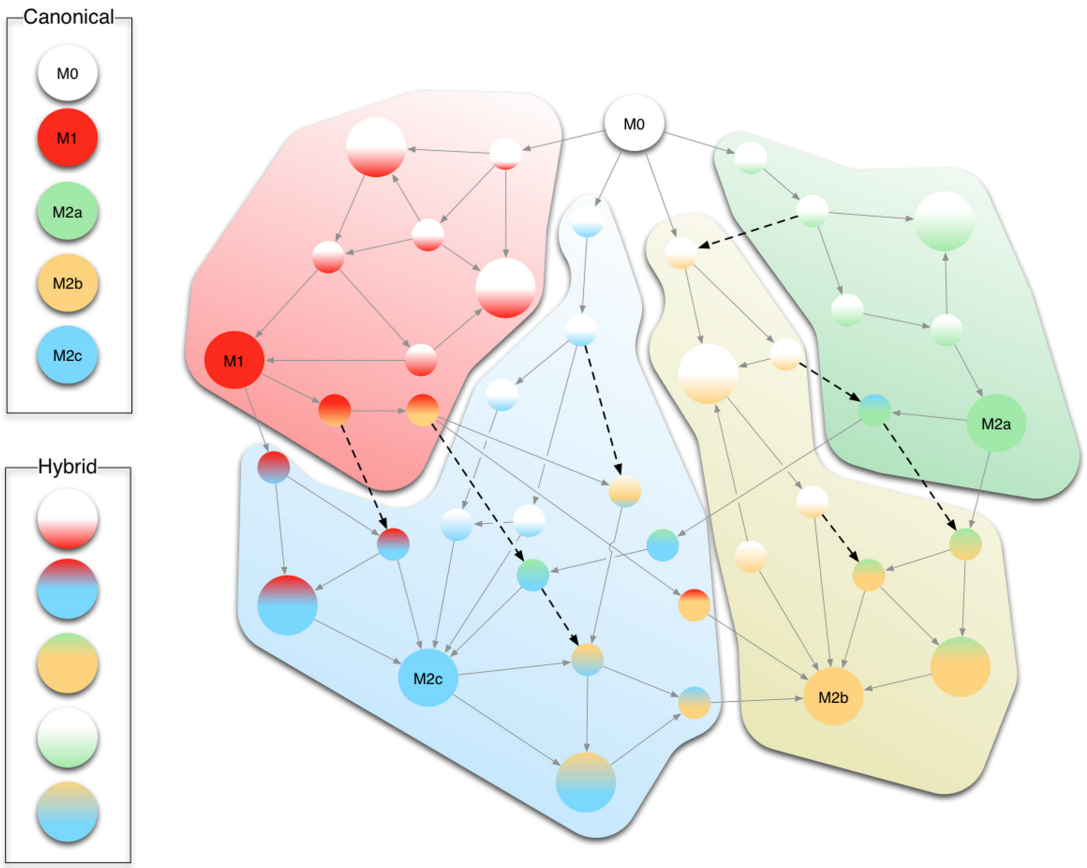


FIGURE 9 | Conceptual representation of the continuum of differentiation states. Circles show intermediate stable states (smaller circles) between the five canonical M0, M1, and M2a/b/c (larger monochromatic circles). Stable states whose correspondent phenotype is not uniquely determined are indicated as larger circles with more than one color. Gray arrows indicate state changes the cell undergoes upon reception of extracellular stimuli. Black dashed arrows show jumps from one differentiation pathway to another. For instance, just by changing the extracellular stimuli (e.g., IL10) a macrophage which started the differentiation from M1 to M2b can divert toward the M2c phenotype.

For instance, in M2-related knockouts *in silico* experiments, such regulators, as for example PPARG, are responsible for the resolution of inflammation and the maintenance of an anti-inflammatory environment by enabling the production of IL-10 and other important anti-inflammatory mediators. Similar studies could focus on networks that are specific to some pathogen or some physiological mechanism, to get a better comprehension in terms of the logic of the regulatory machinery.

This modeling study yielded another important observation, which is related to the environmental-dependent expression of mixed markers identifying one of the four canonical macrophage polarizations. Indeed, recent studies support the view that fully polarized macrophages (M1 and M2) as being the extremes of a continuum of macrophages polarization (Mantovani, 2008). This could for example be obtained by mixing various stimuli, such as IC together with LPS or IL-1β and IL-10, which give rise to M2-like functional phenotypes, yet sharing properties with IL-4-activated macrophages (Mantovani et al., 2004). This continuum of macrophages phenotypes parallels a continuum in CD4+ T cell states, recently observed, as opposed to a limited number of discrete phenotypes (Eizenberg-Magar et al., 2017). Indeed, while T helper cell induction requires the participation of macrophages, several signal feedback mechanisms are implemented for the activation and differentiation of macrophages. Even if this intertwinement may vary in both quantitative and qualitative aspects, the continuum of states detected in T helper and macrophage cells may be more linked than observed up to now.

We surmise that shifts among different phenotypes in our model mimic the hypothetical continuum of macrophage polarization, being M1 and the three subtypes of M2 the extremes of such uninterrupted sequences of states. **Figure 9** conceptualizes this continuum in the progression of gene activations leading from one form of polarization to another driven by various stimuli. For instance, an M1+M2 successive stimuli can lead to an M2a stable configuration while passing through an M1 state (see **Figure 9**).

The presented approach, although promising and general, is not free of pitfalls. Even if little mathematical knowledge is needed to build a Boolean network, the information gained from its analysis is strongly affected by the accuracy of

the relationships among genes encoded in the Boolean rules characterizing the overall dynamics. Manually curated networks optimally convey the biological information but cannot ensure completeness. The usefulness of Boolean networks therefore is found while dealing with poorly characterized systems, especially when quantitative experimental data is missing. In some cases, alternative approaches should be considered such as introducing uncertainty with probabilistic networks or using continuous models that describe the kinetic with greater accuracy than Boolean networks.

To conclude, although there is a wealth of information about the different macrophage subsets *in vitro*, features such as plasticity, heterogeneity, and adaptability make them very difficult to study using conventional experimental tools. In this paper, we have shown that relatively simple logical description of the gene regulation machinery can support the analysis of the emerging complexity of the phenomena of mammalian cell differentiation and can be used to provide testable predictions as, for instance, which combination of stimuli leads to hybrid phenotypes.

The network provided here is manually curated and has been built based on the available information derived from literature to date. This should be considered as-is, that is, limited to the current knowledge which, regarding the less characterized pathways and molecular interactions leading to M2b macrophages, is admittedly lacking.

AUTHOR CONTRIBUTIONS

All authors conceived the study. AP and FC performed the experiments. All authors carried out the analysis and contributed to writing the paper.

ACKNOWLEDGMENTS

GC acknowledges partial support from the European Research Council (grant DEPTH, grant agreement no. 322749) and from the Italian Association for Cancer Research AIRC (grant IG 2017, StateplaceId. 20322). FC and PT acknowledge partial support from the European Commission under the Seventh Framework Programme (MISSION-T2D project, contract no. 600803), and from COST Action CA15120 Open Multiscale Systems Medicine (OpenMultiMed).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphys.2018.01659/full#supplementary-material

REFERENCES

Abou-Jaoudé, W., Monteiro, P. T., Naldi, A., Grandclaudon, M., Soumelis, V., Chaouiya, C., et al. (2014). Model checking to assess T-helper cell plasticity. *Front. Bioeng. Biotechnol.* 2:86. doi: 10.3389/fbioe.2014.00086

Abou-Jaoudé, W., Traynard, P., Monteiro, P. T., Saez-Rodriguez, J., Helikar, T., Thieffry, D., et al. (2016). Logical modeling and dynamical analysis of cellular networks. *Front. Genet.* 7:94. doi: 10.3389/fgene.2016.00094

Abrahams, V. M., Cambridge, G., Lydyard, P. M., and Edwards, J. C. (2000). Induction of tumor necrosis factor alpha production by adhered human monocytes: a key role for Fcgamma receptor type IIIa in rheumatoid arthritis. *Arthritis Rheum.* 43, 608–616. doi: 10.1002/1529-0131(200003)43:3<608::AID-ANR18>3.0.CO;2-G

Albert, I., Thakar, J., Li, S., Zhang, R., and Albert, R. (2008). Boolean network simulations for life scientists. *Source Code Biol. Med.* 3:16. doi: 10.1186/1751-0473-3-16

Arnold, C. E., Whyte, C. S., Gordon, P., Barker, R. N., Rees, A. J., and Wilson, H. M. (2014). A critical role for suppressor of cytokine signalling 3 in promoting M1 macrophage activation and function in vitro and in vivo. *Immunology* 141, 96–110. doi: 10.1111/imm.12173

Baker, B. J., Akhtar, L. N., and Benveniste, E. N. (2009). SOCS1 and SOCS3 in the control of CNS immunity. *Trends Immunol.* 30, 392–400. doi: 10.1016/j.it.2009.07.001

Bally, A. P. R., Lu, P., Tang, Y., Austin, J. W., Scharer, C. D., Ahmed, R., et al. (2015). NF-κB regulates PD-1 expression in macrophages. *J. Immunol.* 194, 4545–4554. doi: 10.4049/jimmunol.1402550

Barahmand-Pour, F., Meinke, A., Groner, B., and Decker, T. (1998). Jak2-Stat5 interactions analyzed in yeast. *J. Biol. Chem.* 273, 12567–12575. doi: 10.1074/jbc.273.20.12567

Benítez, M., and Hejátko, J. (2013). Dynamics of cell-fate determination and patterning in the vascular bundles of *Arabidopsis thaliana,* Candela H, editor. *PLoS One* 8:e63108. doi: 10.1371/journal.pone.0063108

Biswas, S. K., Gangi, L., Paul, S., Schioppa, T., Saccani, A., Sironi, M., et al. (2006). A distinct and unique transcriptional program expressed by A distinct and unique transcriptional program expressed by tumor-associated macrophages (defective NF- B and enhanced IRF-3/STAT1 activation). *Blood* 107, 2112–2122. doi: 10.1182/blood-2005-01-0428

Biswas, S. K., and Mantovani, A. (2010). Macrophage plasticity and interaction with lymphocyte subsets: cancer as a paradigm. *Nat. Immunol.* 11, 889–896. doi: 10.1038/ni.1937

Bouhlel, M. A., Derudas, B., Rigamonti, E., Dièvart, R., Brozek, J., Haulon, S., et al. (2007). PPARγ activation primes human monocytes into alternative M2 macrophages with anti-inflammatory properties. *Cell Metab.* 6, 137–143. doi: 10.1016/j.cmet.2007.06.010

Bowdish, D. M. E., Loffredo, M. S., Mukhopadhyay, S., Mantovani, A., and Gordon, S. (2007). Macrophage receptors implicated in the "adaptive" form of innate immunity. *Microbes Infect.* 9, 1680–1687. doi: 10.1016/j.micinf.09.002

Carbo, A., Hontecillas, R., Andrew, T., Eden, K., Mei, Y., Hoops, S., et al. (2014). Computational modeling of heterogeneity and function of CD4+ T cells. *Front. Cell Dev. Biol.* 2:31. doi: 10.3389/fcell.2014.00031

Castiglione, F., and Celada, F. (2015). *Immune System Modelling and Simulation*. Boca Raton, FL: CRC Press. doi: 10.1201/b18274

Castiglione, F., Tieri, P., Palma, A., and Jarrah, A. S. (2016). Statistical ensemble of gene regulatory networks of macrophage differentiation. *BMC Bioinform.* 17:506. doi: 10.1186/s12859-016-1363-4

Chakraborty, A. K. (2017). A perspective on the role of computational models in immunology. *Annu. Rev. Immunol.* 35, 403–439. doi: 10.1146/annurev-immunol-041015-055325

Chaouiya, C., Naldi, A., and Thieffry, D. (2012). Logical modelling of gene regulatory networks with GINsim. *Methods Mol. Biol.* 804, 463–479. doi: 10.1007/978-1-61779-361-5_23

Chawla, A. (2010). Control of macrophage activation and function by PPARs. *Circ. Res.* 106, 1559–1569. doi: 10.1161/CIRCRESAHA.110.216523

Clatworthy, M. R., Harford, S. K., Mathews, R. J., and Smith, K. G. C. (2014). FcγRIIb inhibits immune complex-induced VEGF-A production and intranodal lymphangiogenesis. *Proc. Natl. Acad. Sci. U.S.A.* 111, 17971–17976. doi: 10.1073/pnas.1413915111

Darnell, J. E., Kerr, I. M., and Stark, G. R. (1994). Jak-STAT pathways and transcriptional activation in response to IFNs and other extracellular signaling proteins. *Science* 264, 1415–1421. doi: 10.1126/science.8197455

Davis, M. J., Tsang, T. M., Qiu, Y., Dayrit, J. K., Freij, J. B., Huffnagle, G. B., et al. (2013). Macrophage M1/M2 polarization dynamically adapts to changes in cytokine microenvironments in *Cryptococcus neoformans* Infection. *MBio.* 4:e264-13. doi: 10.1128/mBio.00264-13

Deng, B., Wehling-Henricks, M., Villalta, A. A., Wang, Y., and Tidball, J. G. (2012). Interleukin-10 triggers changes in macrophage phenotype that promote muscle growth and regeneration. *J. Immunol.* 189, 53669–53680. doi: 10.1038/nmeth.2250.Digestion

Dickensheets, H., Vazquez, N., Sheikh, F., Gingras, S., Murray, P. J., Ryan, J. J., et al. (2007). Suppressor of cytokine signaling-1 is an IL-4-inducible gene in macrophages and feedback inhibits IL-4 signaling. *Genes Immun.* 8, 21–27. doi: 10.1038/sj.gene.6364352

Doyle, S., Vaidya, S., O'Connell, R., Dadgostar, H., Dempsey, P., Wu, T., et al. (2002). IRF3 mediates a TLR3/TLR4-specific antiviral gene program. *Immunity* 17, 251–63. doi: 10.1016/S1074-7613(02)00390-4

Eftimie, R., Gillard, J. J., and Cantrell, D. A. (2016). Mathematical models for immunology: current state of the art and future research directions. *Bull. Math. Biol.* 78, 2091–2134. doi: 10.1007/s11538-016-0214-9

Eizenberg-Magar, I., Rimer, J., Zaretsky, I., Lara-Astiaso, D., Reich-Zeliger, S., and Friedman, N. (2017). Diverse continuum of CD4 + T-cell states is determined by hierarchical additive integration of cytokine signals. *Proc. Natl. Acad. Sci. U.S.A.* 114, E6447–E6456. doi: 10.1073/pnas.1615590114

Espinosa-Soto, C., Padilla-Longoria, P., and Alvarez-Buylla, E. R. (2004). A Gene regulatory network model for cell-fate determination during *Arabidopsis thaliana* flower development that is robust and recovers experimental gene expression profiles. *Plant Cell.* 16, 2923–2939. doi: 10.1105/tpc.104.021725

Foey, A. D. (2014). *Macrophages—Masters of Immune Activation, Suppression and Deviation, Immune Response Activation, Guy Huynh Thien Duc*, Chap. 5. London: IntechOpen, 121–149. doi: 10.5772/57541

Gao, C.-H., Dong, H.-L., Tai, L., and Gao, X.-M. (2018). Lactoferrin-containing immunocomplexes drive the conversion of human macrophages from M2- into M1-like phenotype. *Front. Immunol.* 9:37. doi: 10.3389/fimmu.2018.00037

Gong, D., Shi, W., Yi, S., Chen, H., Groffen, J., and Heisterkamp, N. (2012). TGFβ signaling plays a critical role in promoting alternative macrophage activation. *BMC Immunol.* 13:31. doi: 10.1186/1471-2172-13-31

Gordon, S. (2003). Alternative activation of macrophages. *Nat. Rev. Immunol.* 3, 23–35. doi: 10.1038/nri978

Gordon, S. (2008). Elie metchnikoff: father of natural immunity. *Eur. J. Immunol.* 38, 3257–3264. doi: 10.1002/eji.200838855

Gordon, S., and Martinez, F. O. (2010). Alternative activation of macrophages: mechanism and functions. *Immunity* 32, 593–604. doi: 10.1016/j.immuni.2010.05.007

Gordon, S., Plüddemann, A., and Martinez Estrada, F. (2014). Macrophage heterogeneity in tissues: phenotypic diversity and functions. *Immunol. Rev.* 262, 36–55. doi: 10.1111/imr.12223

Guevara, M. R. (2003). *Bifurcations Involving Fixed Points and Limit Cycles in Biological Systems*. New York, NY: Springer, 41–85. doi: 10.1007/978-0-387-21640-9_3

Guilliams, M., Bruhns, P., Saeys, Y., Hammad, H., and Lambrecht, B. N. (2014). The function of Fcγ receptors in dendritic cells and macrophages. *Nat. Rev. Immunol. Nat.* 14, 94–108. doi: 10.1038/nri3582

Hamilton, J. A. (2008). Colony-stimulating factors in inflammation and autoimmunity. *Nat. Rev. Immunol.* 8, 533–545. doi: 10.1016/S1471-4906(02)02260-3

Hirano, M., Davis, R. S., Fine, W. D., Nakamura, S., Shimizu, K., Yagi, H., et al. (2007). IgEb immune complexes activate macrophages through FcgammaRIV binding. *Nat. Immunol.* 8, 762–771. doi: 10.1038/ni1477

Honda, K., Yanai, H., Takaoka, A., and Taniguchi, T. (2005). Regulation of the type I IFN induction: a current view. *Int. Immunol.* 17, 1367–1378. doi: 10.1093/intimm/dxh318

Hutchins, A. P., Diez, D., and Miranda-Saavedra, D. (2013). The IL-10/STAT3-mediated anti-inflammatory response: recent developments and future challenges. *Brief Funct. Genomics* 12, 489–498. doi: 10.1093/bfgp/elt028

Ishii, M., Wen, H., Corsa, C. A. S., Liu, T., Coelho, A. L., Allen, R. M., et al. (2009). Epigenetic regulation of the alternatively activated macrophage phenotype. *Blood* 114, 3244–3254. doi: 10.1182/blood-2009-04-217620

Ito, S., Ansari, P., Sakatsume, M., Dickensheets, H., Vazquez, N., Donnelly, R. P., et al. (1999). Interleukin-10 inhibits expression of both interferon alpha- and interferon gamma- induced genes by suppressing tyrosine phosphorylation of STAT1. *Blood* 93, 1456–1463.

Ji, J.-D., Tassiulas, I., Park-Min, K.-H., Aydin, A., Mecklenbrauker, I., Tarakhovsky, A., et al. (2003). Inhibition of interleukin 10 signaling after Fc receptor ligation and during rheumatoid arthritis. *J. Exp. Med.* 197, 1573–1583. doi: 10.1084/jem.20021820

Karlebach, G., and Shamir, R. (2008). Modelling and analysis of gene regulatory networks. *Nat. Rev. Mol. Cell. Biol.* 9, 770–780. doi: 10.1038/nrm2503

Kawai, T., and Akira, S. (2010). The role of pattern-recognition receptors in innate immunity: update on Toll-like receptors. *Nat. Immunol.* 11, 373–384. doi: 10.1038/ni.1863

Kawai, T., and Akira, S. (2011). Toll-like receptors and their crosstalk with other innate receptors in infection and immunity. *Immunity* 34, 637–650. doi: 10.1016/j.immuni.2011.05.006

Kawai, T., Takeuchi, O., Fujita, T., Inoue, J., Mühlradt, P. F., Sato, S., et al. (2001). Lipopolysaccharide stimulates the MyD88-independent pathway and results in activation of IFN-regulatory factor 3 and the expression of a subset of lipopolysaccharide-inducible genes. *J. Immunol.* 167, 5887–5894. doi: 10.4049/jimmunol.167.10.5887

Kestler, H. A., Wawra, C., Kracher, B., and Kühl, M. (2008). Network modeling of signal transduction: establishing the global view. *BioEssays* 30, 1110–1125. doi: 10.1002/bies.20834

Kotenko, S.V., Izotova, L. S., Pollack, B. P., Mariano, T. M., Donnelly, R. J., Muthukumaran, G., et al. (1995). Interaction between the components of the interferon gamma receptor complex. *J. Biol. Chem.* 270, 20915–20921. doi: 10.1074/jbc.270.36.20915

Kraakman, M. J., Murphy, A. J., Jandeleit-Dahm, K., and Kammoun, H. L. (2014). Macrophage polarization in obesity and type 2 diabetes: weighing down our understanding of macrophage function? *Front. Immunol.* 5:470. doi: 10.3389/fimmu.2014.00470

Krausgruber, T., Blazek, K., Smallie, T., Alzabin, S., Lockstone, H., Sahgal, N., et al. (2011). IRF5 promotes inflammatory macrophage polarization and TH1-TH17 responses. *Nat. Immunol.* 12, 231–238. doi: 10.1038/ni.1990

Lang, R., Patel, D., Morris, J. J., Rutschman, R. L., and Murray, P. J. (2002a). Shaping gene expression in activated and resting primary macrophages by IL-10. *J. Immunol.* 169, 2253–2263. doi: 10.4049/jimmunol.169.5.2253

Lang, R., Rutschman, R. L., Greaves, D. R., and Murray, P. J. (2002b). Autocrine deactivation of macrophages in transgenic mice constitutively overexpressing IL-10 under control of the human CD68 promoter. *J. Immunol. Am. Assoc. Immunol.* 168, 3402–3411. doi: 10.4049/JIMMUNOL.168.7.3402

Lawrence, T., and Natoli, G. (2011). Transcriptional regulation of macrophage polarization: enabling diversity with identity. *Nat. Rev. Immunol.* 11, 750–761. doi: 10.1038/nri3088

Lehtonen, A., Matikainen, S., Miettinen, M., and Julkunen, I. (2002). Granulocyte-macrophage colony-stimulating factor (GM-CSF)-induced STAT5 activation and target-gene expression during human monocyte/macrophage differentiation. *J. Leukoc Biol.* 71, 511–519.

Liu, W., Ouyang, X., Yang, J., Liu, J., Li, Q., Gu, Y., et al. (2009). AP-1 activated by toll-like receptors regulates expression of IL-23 p19. *J. Biol. Chem.* 284, 24006–24016. doi: 10.1074/jbc.M109.025528

Lucas, M., Zhang, X., Prasanna, V., and Mosser, D. M. (2005). ERK activation following macrophage FcgammaR ligation leads to chromatin modifications at the IL-10 locus. *J. Immunol.* 175, 469–477. doi: 10.4049/jimmunol.175.1.469

Luo, Y., Pollard, J. W., and Casadevall, A. (2010). FC Receptor cross-linking stimulates cell proliferation of macrophages via the ERK pathway. *J. Biol. Chem.* 285, 4232–4242. doi: 10.1074/jbc.M109.037168

Maiti, S., Dai, W., Alaniz, R., Hahn, J., and Jayaraman, A. (2014). Mathematical modeling of pro- and anti-inflammatory signaling in macrophages. *Process. Multidiscipl. Digital Publish. Inst.* 3, 1–18. doi: 10.3390/pr3010001

Mantovani, A. (2008). From phagocyte diversity and activation to probiotics: back to Metchnikoff. *Eur. J. Immunol.* 38, 3269–3273. doi: 10.1002/eji.200838918

Mantovani, A., Sica, A., Sozzani, S., Allavena, P., Vecchi, A., and Locati, M. (2004). The chemokine system in diverse forms of macrophage activation and polarization. *Trends Immunol.* 25, 677–686. doi: 10.1016/j.it.2004.09.015

Mantovani, A., Sozzani, S., Locati, M., Allavena, P., and Sica, A. M. (2002). Macrophage polarization: tumor-associated macrophages as a paradigm for polarized M2 mononuclear phagocytes. *Trends Immunol.* 23, 549–555. doi: 10.1016/S1471-4906(02)02302-5

Mao, A.-P., Shen, J., and Zuo, Z. (2015). Expression and regulation of long noncoding RNAs in TLR4 signaling in mouse macrophages. *BMC Genomics* 16:45. doi: 10.1186/s12864-015-1270-5

Martinez, F. O., Helming, L., and Gordon, S. (2009). Alternative activation of macrophages: an immunologic functional perspective. *Annu. Rev. Immunol.* 27, 451–483. doi: 10.1146/annurev.immunol.021908.132532

Martinez-Sanchez, M. E., Mendoza, L., Villarreal, C., and Alvarez-Buylla, E. R. (2015). A minimal regulatory network of extrinsic and intrinsic factors recovers observed patterns of CD4+ T cell differentiation and plasticity. *PLoS Comput. Biol. Public Libr. Sci.* 11:e1004324. doi: 10.1371/journal.pcbi.1004324

McLaren, J. E., and Ramji, D. P. (2009). Interferon gamma: a master regulator of atherosclerosis. *Cytokine Growth Fact. Rev.* 20, 125–135. doi: 10.1016/j.cytogfr.2008.11.003

Méndez, A., and Mendoza, L. (2016). A network model to describe the terminal differentiation of B cells. *PLoS Comput. Biol.* 12:e1004696. doi: 10.1371/journal.pcbi.1004696

Moore, K. W., de Waal Malefyt, R., Coffman, R. L., and O'Garra, A. (2001). Interleukin-10 and the interleukin-10 receptor. *Annu. Rev. Immunol.* 19, 683–765. doi: 10.1146/annurev.immunol.19.1.683

Mosser, D. M., and Edwards, J. P. (2008). Exploring the full spectrum of macrophage activation. *Nat. Rev. Immunol.* 8, 958–969. doi: 10.1038/nri2448

Müssel, C., Hopfensitz, M., and Kestler, H. A. (2010). BoolNet—an R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics* 26, 1378–1380. doi: 10.1093/bioinformatics/btq124

Nakamura, R., Sene, A., Santeford, A., Gdoura, A., Kubota, S., Zapata, N., et al. (2015). IL10-driven STAT3 signalling in senescent macrophages promotes pathological eye angiogenesis. *Nat. Commun.* 6:7847. doi: 10.1038/ncomms8847

Naldi, A., Berenguier, D., Fauré, A., Lopez, F., Thieffry, D., and Chaouiya, C. (2009). Logical modelling of regulatory networks with GINsim 2.3. *BioSystems* 97, 134–139. doi: 10.1016/j.biosystems.2009.04.008

Naldi, A., Carneiro, J., Chaouiya, C., and Thieffry, D. (2010). Diversity and plasticity of Th cell types predicted from regulatory network modelling, Bonneau R, editor. *PLoS Comput. Biol.* 6:e1000912. doi: 10.1371/journal.pcbi.1000912

Nimmerjahn, F., and Ravetch, J. V. (2008). Fcgamma receptors as regulators of immune responses. *Nat. Rev. Immunol.* 8, 34–47. doi: 10.1038/nri2206

Novak, M., Weinheimer-Haus, E., and Koh, T. (2014). Macrophage activation and skeletal muscle healing following traumatic injury. *J. Pathol.* 232, 344–355. doi: 10.1038/nmeth.2250.Digestion

O'Carroll, C., Fagan, A., Shanahan, F., and Carmody, R. J. (2013). Identification of a unique hybrid macrophage-polarization state following recovery from lipopolysaccharide tolerance. *J. Immunol.* 192, 427–436. doi: 10.4049/jimmunol.1301722

Ortiz-Gutiérrez, E., García-Cruz, K., Azpeitia, E., Castillo, A., Sánchez Mde la, P., and Álvarez-Buylla, E. R. (2015). A dynamic gene regulatory network model that recovers the cyclic behavior of *Arabidopsis thaliana* cell cycle, Albert R, editor. *PLoS Comput. Biol.* 11:e1004486. doi: 10.1371/journal.pcbi.1004486

Park, B. S., Song, D. H., Kim, H. M., Choi, B.-S., Lee, H., and Lee, J.-O. (2009). The structural basis of lipopolysaccharide recognition by the TLR4-MD-2 complex. *Nature* 458, 1191–1195. doi: 10.1038/nature07830

Perfetto, L., Briganti, L., Calderone, A., Perpetuini, A. C., Iannuccelli, M., Langone, F., et al. (2016). SIGNOR: a database of causal relationships between biological entities. *Nucleic Acids Res.* 44, D548–D554. doi: 10.1093/nar/gkv1048

Platanias, L. C. (2005). Mechanisms of type-I- and type-II-interferon-mediated signalling. *Nat. Rev. Immunol.* 5, 375–386. doi: 10.1038/nri1604

Raes, G., Brys, L., Dahal, B. K., Brandt, J., Grooten, J., Brombacher, F., et al. (2005). Macrophage galactose-type C-type lectins as novel markers for alternatively activated macrophages elicited by parasitic infections and allergic airway inflammation. *J. Leukoc Biol.* 77, 321–327. doi: 10.1189/jlb.0304212

Rauch, I., Müller, M., and Decker, T. (2013). The regulation of inflammation by interferons and their STATs. *JAKSTAT* 2:e23820. doi: 10.4161/jkst.23820

Ricote, M., Li, A. C., Willson, T. M., Kelly, C. J., and Glass, C. K. (1998). The peroxisome proliferator-activated receptor-gamma is a negative regulator of macrophage activation. *Nature* 391, 79–82. doi: 10.1038/34178

Rigamonti, E., Zordan, P., Sciorati, C., Rovere-querini, P., and Brunelli, S. (2014). Macrophage plasticity in skeletal muscle repair. *Biomed. Res. Int.* 2014:560629. doi: 10.1155/2014/560629

Riley, J. K., Takeda, K., Akira, S., and Schreiber, R. D. (1999). Interleukin-10 receptor signaling through the JAK-STAT pathway. *J. Biol. Chem.* 274, 16513–16521. doi: 10.1074/jbc.274.23.16513

Ritter, M., Buechler, C., Langmann, T., Orso, E., Klucken, J., and Schmitz, G. (1999). The scavenger receptor CD 163: regulation, promoter structure and genomic organization. *Pathobiology* 67, 257–261. doi: 10.1159/000028105

Sadler, A. J., and Williams, B. R. G. (2008). Interferon-inducible antiviral effectors. *Nat. Rev. Immunol.* 8, 559–568. doi: 10.1038/nri2314

Samal, A., and Jain, S. (2008). The regulatory network of *E. coli* metabolism as a Boolean dynamical system exhibits both homeostasis and flexibility of response. *BMC Syst. Biol.* 2:21. doi: 10.1186/1752-0509-2-21

Sánchez-Mejorada, G., and Rosales, C. (1998). Signal transduction by immunoglobulin Fc receptors. *J. Leukoc Biol.* 63, 521–533. doi: 10.1002/jlb.63.5.521

Sang, Y., Miller, L. C., and Blecha, F. (2015). Macrophage polarization in virus-host interactions. *J. Clin. Cell Immunol.* 6:311. doi: 10.4172/2155-9899.1000311

Sanin, D. E., Prendergast, C. T., and Mountford, A. P. (2015). IL-10 production in macrophages is regulated by a TLR-Driven CREB-mediated mechanism that is linked to genes involved in cell metabolism. *J. Immunol.* 195, 1218–1232. doi: 10.4049/jimmunol.1500146

Santoni, D., Pedicini, M., and Castiglione, F. (2008). Implementation of a regulatory gene network to simulate the TH1/2 differentiation in an agent-based model of hypersensitivity reactions. *Bioinformatics* 24, 1374–1380. doi: 10.1093/bioinformatics/btn135

Sato, A., Ohtaki, H., Tsumuraya, T., Song, D., Ohara, K., Asano, M., et al. (2012). Interleukin-1 participates in the classical and alternative activation of microglia/macrophages after spinal cord injury. *J. Neuroinflamm.* 9:553. doi: 10.1186/1742-2094-9-65

Satoh, T., Takeuchi, O., Vandenbon, A., Yasuda, K., Tanaka, Y., Kumagai, Y., et al. (2010). The Jmjd3-Irf4 axis regulates M2 macrophage polarization and host responses against helminth infection. *Nat. Immunol.* 11, 936–944. doi: 10.1038/ni.1920

Scandlyn, J. (2000). When AIDS became a chronic disease. *West J Med.* 172, 130–133. doi: 10.1136/ewjm.172.2.130

Sheikh, F., Dickensheets, H., Gamero, A. M., Vogel, S. N., and Donnelly, R. P. (2014). An essential role for IFN-β in the induction of IFN-stimulated gene expression by LPS in macrophages. *J. Leukoc Biol.* 96, 591–600. doi: 10.1189/jlb.2A0414-191R

Sica, A., and Mantovani, A. (2012). Macrophage plasticity and polarization: in vivo veritas. *J. Clin. Invest.* 122, 787–795. doi: 10.1172/JCI59643

Sutterwala, F. S., Noel, G. J., Salgame, P., and Mosser, D. M. (1998). Reversal of proinflammatory responses by ligating the macrophage Fcgamma receptor type I. *J. Exp. Med.* 188, 217–222. doi: 10.1084/jem.188.1.217

Thomas, R., and Kaufman, M. (2001). Multistationarity, the basis of cell differentiation and memory. I. Structural conditions of multistationarity and other nontrivial behavior. *Chaos* 11:170. doi: 10.1063/1.1350439

Thomas, R., and D'Ari, R. (1990). *Biological Feedback*. Boca Raton, FL:CRC Press.

Tieri, P., Prana, V., Colombo, T., Santoni, D., and Castiglione, F. (2014). Multi-scale simulation of T helper lymphocyte differentiation. *Adv. Bioinform. Comput. Biol.* 8826, 123–134. doi: 10.1007/978-3-319-12418-6_16

Tran-Thi, T. A., Decker, K., and Baeuerle, P. A. (1995). Differential activation of transcription factors NF-kappa B and AP-1 in rat liver macrophages. *Hepatology* 22, 613–619. doi: 10.1002/hep.1840220235

Vogelpoel, L. T. C., Baeten, D. L. P., de Jong, E. C., and den Dunnen, J. (2015). Control of cytokine production by human Fc gamma receptors: implications for pathogen defense and autoimmunity. *Front. Immunol.* 6:79. doi: 10.3389/fimmu.2015.00079

Vogelpoel, L. T. C., Hansen, I. S., Rispens, T., Muller, F. J. M., van Capel, T. M. M., Turina, M. C., et al. (2014). Fc gamma receptor-TLR cross-talk elicits pro-inflammatory cytokine production by human M2 macrophages. *Nat. Commun.* 5:5444. doi: 10.1038/ncomms6444

Wang, N., Liang, H., and Zen, K. (2014). Molecular mechanisms that influence the macrophage M1-M2 polarization balance. *Front. Immunol.* 5:614. doi: 10.3389/fimmu.2014.00614

Weber, A., Wasiliew, P., and Kracht, M. (2010). Interleukin-1 (IL-1) pathway. *Sci. Signal.* 3:cm1. doi: 10.1126/scisignal.3105cm1

Whyte, C. S., Bishop, E. T., Ruckerl, D., Gaspar-Pereira, S., Barker, R. N., Allen, J. E., et al. (2011). Suppressor of cytokine signaling (SOCS)1 is a key determinant of differential macrophage activation and function. *J. Leukoc Biol.* 90, 845–854. doi: 10.1189/jlb.1110644

Yamaoka, K., Otsuka, T., Niiro, H., Arinobu, Y., Niho, Y., Hamasaki, N., et al. (1998). Activation of STAT5 by lipopolysaccharide through granulocyte-macrophage colony-stimulating factor production in human monocytes. *J. Immunol.* 160, 838–845.

Zhang, Y.-H., He, M., Wang, Y., and Liao, A.-H. (2017). Modulators of the Balance between M1 and M2 macrophages during pregnancy. *Front. Immunol.* 8:120. doi: 10.3389/fimmu.2017.00120

Zhang, Y., Liu, S., Liu, J., Zhang, T., Shen, Q., Yu, Y., et al. (2009). Immune complex/Ig negatively regulate TLR4-triggered inflammatory response in macrophages through Fc gamma RIIb-dependent PGE2 production. *J. Immunol.* 182, 554–562. doi: 10.4049/jimmunol.182.1.554

Zheng, X.-F., Hong, Y.-X., Feng, G.-J., Zhang, G.-F., Rogers, H., Lewis, M. A. O., et al. (2013). Lipopolysaccharide-Induced M2 to M1 macrophage transformation for IL-12p70 production is blocked by candida albicans mediated up-regulation of EBI3 expression, Tran DQ, editor. *PLoS One* 8:e63967. doi: 10.1371/journal.pone.0063967

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

ORIGINAL RESEARCH
published: 20 December 2018
doi: 10.3389/fphys.2018.01836

Modeling the Role of the Microbiome in Evolution

Saúl Huitzil[1], *Santiago Sandoval-Motta*[2,3,4], *Alejandro Frank*[2,5,6] *and Maximino Aldana*[1,2]*

[1] *Instituto de Ciencias Físicas, Universidad Nacional Autónoma de México, Cuernavaca, Mexico,* [2] *Centro de Ciencias de la Complejidad, Universidad Nacional Autónoma de México, Mexico City, Mexico,* [3] *Instituto Nacional de Medicina Genómica, Mexico City, Mexico,* [4] *Consejo Nacional de Ciencia y Tecnología, Cátedras CONACyT, Mexico City, Mexico,* [5] *Instituto de Ciencias Nucleares, Universidad Nacional Autónoma de México, Mexico City, Mexico,* [6] *Member of El Colegio Nacional, Mexico City, Mexico*

There is undeniable evidence showing that bacteria have strongly influenced the evolution and biological functions of multicellular organisms. It has been hypothesized that many host-microbial interactions have emerged so as to increase the adaptive fitness of the *holobiont* (the host plus its microbiota). Although this association has been corroborated for many specific cases, general mechanisms explaining the role of the microbiota in the evolution of the host are yet to be understood. Here we present an evolutionary model in which a network representing the host adapts in order to perform a predefined function. During its adaptation, the host network (HN) can interact with other networks representing its microbiota. We show that this interaction greatly accelerates and improves the adaptability of the HN without decreasing the adaptation of the microbial networks. Furthermore, the adaptation of the HN to perform several functions is possible only when it interacts with many different bacterial networks in a specialized way (each bacterial network participating in the adaptation of one function). Disrupting these interactions often leads to non-adaptive states, reminiscent of dysbiosis, where none of the networks the holobiont consists of can perform their respective functions. By considering the holobiont as a unit of selection and focusing on the adaptation of the host to predefined but arbitrary functions, our model predicts the need for specialized diversity in the microbiota. This structural and dynamical complexity in the holobiont facilitates its adaptation, whereas a homogeneous (non-specialized) microbiota is inconsequential or even detrimental to the holobiont's evolution. To our knowledge, this is the first model in which symbiotic interactions, diversity, specialization and dysbiosis in an ecosystem emerge as a result of coevolution. It also helps us understand the emergence of complex organisms, as they adapt more easily to perform multiple tasks than non-complex ones.

Keywords: holobiont, coevolution, microbiome, symbiosis, complex networks, adaptability, microbiota diversity

OPEN ACCESS

Edited by:
Matteo Barberis,
University of Amsterdam, Netherlands

Reviewed by:
Reka Albert,
Pennsylvania State University, United States
Brian Paul Ingalls,
University of Waterloo, Canada

***Correspondence:**
Maximino Aldana
max@icf.unam.mx

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: *08 August 2018*
Accepted: *06 December 2018*
Published: *20 December 2018*

Citation:
Huitzil S, Sandoval-Motta S, Frank A and Aldana M (2018) Modeling the Role of the Microbiome in Evolution. Front. Physiol. 9:1836. doi: 10.3389/fphys.2018.01836

INTRODUCTION

It has been firmly established during the last decade that the microbiota of a multicellular host strongly influences its evolution and adaptation (Ley et al., 2008; Zilber-Rosenberg and Rosenberg, 2008; Rosenberg et al., 2010; Brucker and Bordenstein, 2013; Andrew et al., 2016; Rosenberg and Zilber-Rosenberg, 2016; Sharpton, 2018). In turn, the host's ability to interact with other organisms and modify its environment to its advantage can guide the composition of its microbiota (Mai, 2004; Spor et al., 2011; Yatsunenko et al., 2012; Moeller et al., 2014). For instance, the human

microbiota plays an important role in many fundamental physiological functions, such as the development of the immune system (Hooper et al., 2012), degradation of fiber and metabolization of fats and carbohydrates (Krajmalnik-Brown et al., 2012), regulation of bone density (McCabe et al., 2015), metabolization of drugs (Wilson and Nicholson, 2017) and control of infections by pernicious bacteria like *Clostridium difficile* (Rupnik et al., 2009; Van Nood et al., 2013; Seekatz and Young, 2014). A state of imbalance in the human microbiota, known as dysbiosis, has been correlated with diseases (Cho and Blaser, 2012) such as obesity (Ley et al., 2006b; Ley, 2010), inflammatory bowel disease (Morgan et al., 2012; Halfvarson et al., 2017), cancer (Farrell et al., 2011; Zackular et al., 2013; Francescone et al., 2014; Sears and Garrett, 2014; Contreras et al., 2016; Yang et al., 2017) and even neurological disorders as schizophrenia and autism (Gonzalez et al., 2011; Rogers et al., 2016). The system consisting of the host and its microbiota, known as *holobiont*, exhibits the unequivocal existence of symbiotic relationships between microbes and multicellular organisms (Theis et al., 2016). Given these complex interdependencies, the holobiont has been proposed to function as a single evolutionary unit (Gilbert et al., 2012; Gordon et al., 2013; Guerrero et al., 2013; Bordenstein and Theis, 2015; Theis et al., 2016; Roughgarden et al., 2018). This is because environmental changes may impose selective pressures on the host which in turn will affect its microbiota (for a recent review see Roughgarden et al., 2018). In the light of these findings, it has been suggested that evolutionary theories have to be either reformulated or expanded in order to account for the adaptability of the holobiont as an evolutionary unit (Laland et al., 2014; Ereshefsky and Pedroso, 2015; Van Opstal and Bordenstein, 2015; Sandoval-Motta et al., 2017). Whether the holobiont is or is not an evolutionary unit is still a matter of debate (Moran and Sloan, 2015; Douglas and Werren, 2016; Doolittle and Booth, 2017; Doolittle and Inkpen, 2018). However, here we show that selective pressures applied to the host and its associated microbes taken as whole, can help us explain how symbiotic relationships in holobionts arose and are currently maintained.

Examples of the influence that microorganisms have had on the adaptation of their hosts range from cases in which microbes help the host to perform specific non-essential functions, to cases in which microbes have completely substituted essential functions of the host (Sagan, 1967; Zilber-Rosenberg and Rosenberg, 2008; Queller and Strassmann, 2016; Roughgarden et al., 2018). Nevertheless, the specific mechanisms by which this influence is carried on are not yet known. Particularly, what are the general benefits that the microbiota provides to the host during its evolution is still an open question. A possible answer to it is that the adaptation time of the host to face new environmental challenges is considerably reduced due to the great diversity and plasticity of its microbiota (Zilber-Rosenberg and Rosenberg, 2008; Rosenberg and Zilber-Rosenberg, 2016). This hypothesis assumes that the emergence of strong symbiotic relationships between the host and its microbiota occurs at the genetic and metabolic levels, for only in this way changes occurring in the microbiota can rapidly propagate to the host's metabolism and affect its adaptability. Indeed, recent evidence shows that the microbiota can regulate metabolic pathways and gene expression patterns of its host, and due to this interaction the host can properly perform cell differentiation, tissue formation, nutrition and other important functions (Hooper et al., 2001; Rawls et al., 2004; Bates et al., 2006; Shin et al., 2011; Nicholson et al., 2012; Camp et al., 2014).

It has been proposed that natural selection operating at the Host-level promotes stable and redundant microbial societies, whereas selection operating at the microbial level promotes functional specialization of their component species (Ley et al., 2006a). Despite all the knowledge we have now on human associated microbial communities, we still do not fully understand the evolutionary forces behind the diversity observed in our microbiota. On the one hand, the most abundant ecological relationship between microbial species is competition (Foster and Bell, 2012; Coyte et al., 2015; Moran and Sloan, 2015; Douglas and Werren, 2016), which often leads to uniform microbial communities where just a few species dominate the whole environment. On the other hand, it has been shown that purely mutualistic interactions lead to unstable communities as their diversity increases. These observations are at odds with the great diversity and stability observed in the microbiota of most plants and animals. Maintaining this diversity is fundamental for the survivability of the host, as it is known that a loss in the microbiota's diversity may produce severe dysbiosis that can result in host diseases or even death (Blaser and Falkow, 2009; Turnbaugh and Stintzi, 2011; Cho and Blaser, 2012; Fernández et al., 2013; Lloyd-Price et al., 2016; Blaser, 2017). An observation that circumvents this caveat is that multicellular organisms have developed different mechanisms to maintain the equilibrium between its diverse microbial communities. These mechanisms tend to compartmentalize the microbes in separate niches while reducing the interactions between microbes in the same niche (Grice and Segre, 2011; Donaldson et al., 2015; Deines et al., 2017; Tropini et al., 2017; Roughgarden et al., 2018). Understanding why microbial diversity is necessary for the evolution and adaptation of the host, and why disease arises when such diversity is lost, is a fundamental question with still no definitive answer.

To address these questions, we adopt the hypothesis that the holobiont constitutes a unit of selection in evolution and explore its consequences. We present an evolutionary population model in which the biological functions of organisms are encoded in the Boolean dynamics of regulatory networks. In our model, a host is represented as a Boolean network that needs to evolve in order to adequately perform a predefined task (or function). This is equivalent to the host acquiring a new phenotype in order to cope with a new environmental challenge. A population of such host networks is evolved in a way that each host network can establish regulatory interactions with a set of microbial species, each one represented also by a network. The main difference between the microbial and host networks is that due to the faster duplication rates of microbes, the generation of mutants is at least one order of magnitude larger in the microbial networks than in the host. Mutants, as explained in detail in the Materials and Methods section (M&M), are simulated by rewiring the connections of their network, or by altering their functionality. As we are dealing with

evolutionary dynamics, it is important to mention that we will only consider host-microbe interactions that can be transmitted across generations. This is based on the fact that in many species, parents directly transmit their microbiota to their offspring or they construct environments with a stable microbial composition that bias the microbial composition of their progeny (Rosenberg et al., 2010; Fitzpatrick, 2014). Another important assumption in our model is the persistence across generations of the host-microbe interactions developed throughout the evolution of the holobiont, which is a necessary condition for natural selection to operate (Doolittle and Booth, 2017). Additionally, we implement the "It's the Song not the Singer" approach proposed by Doolittle (Taxis et al., 2015; Doolittle and Booth, 2017; Doolittle and Inkpen, 2018) by preserving throughout the evolution of the holobiont, those regulatory connections that contribute to the host's adaptation to perform a predefined but otherwise arbitrary dynamical function. The conservation of the dynamical function across generations occurs regardless of the specific host-microbe network interactions that are contributing to the adaptation process.

Our evolutionary model is based on the Boolean network model introduced by S. Kauffman (presented in the M&M section) to describe gene regulation and cell differentiation processes (Kauffman, 1969a,b). During the last 20 years, it has been shown that this model adequately captures the main aspects of gene regulation dynamics. For instance, Boolean networks are able to reproduce gene expression patterns and metabolic pathways experimentally observed in organisms such as *Arabidopsis thaliana* (Espinosa-Soto et al., 2004), *Drosophila melanogaster* (Albert and Othmer, 2003), yeast (Li et al., 2004; Davidich and Bornholdt, 2008), human epithelial cells (Huang et al., 2005) and murine blood progenitor cells (Hameya et al., 2017) among others. Additionally, Huang et al. experimentally showed that the dynamical attractors of a Boolean network correspond to different cell types or cell fates (Huang et al., 2005). Because of this evidence, we use Boolean networks to represent the gene regulation networks of both the hosts and their microbes. Since we are interested in general principles about the emergence of symbiotic interactions, we use random networks instead of carefully constructed ones corresponding to specific organisms. Although the gene regulatory network of an organism greatly determines its phenotype (Davidson and Levine, 2008; Oliveri et al., 2008), it is known that several functions depend more on the general structure of the network than on the specific genes involved (Wagner, 2007). Therefore, using random Boolean networks in our population model has the advantage of determining the capability of the network to acquire new functions throughout its evolution regardless of its detailed composition (Davidson, 2010). This function-centered approach is consistent with the fact that a core microbiome is more likely to be identified based on functionalities rather than on the particular phylogenetic details of its species (Consortium, 2012; Taxis et al., 2015; Doolittle and Booth, 2017; Doolittle and Inkpen, 2018). We describe in detail the Boolean network model in M&M section.

Simulations of this evolutionary model show that the adaptation of the host network is greatly enhanced when it interacts with the microbial networks, which are the ones that absorb most of the mutations without changing their own adaptation. Additionally, the host network can improve its adaptation to perform multiple functions only if the set of microbial networks is partitioned into specialized subsets (niches), each one participating in the host's adaptation to a small number of functions. This specialization provides the holobiont with a structural and dynamical complexity that facilitates its evolution, whereas non-specialized microbiota is shown to be either inconsequential or detrimental to the holobiont's adaptation. Once the holobiont is adapted, the disconnection of one or more of these specialized niches leads a global incompetence to perform the required set of imposed tasks. This is reminiscent of the dysbiosis observed in real organisms when their microbiota's diversity is reduced. To our knowledge, this is the first model in which symbiotic interactions, diversity, specialization and dysbiosis in an ecosystem emerge as a result of coevolution. It also helps us understand the emergence of complex organisms, as they adapt more easily than unstructured ones.

MODEL AND RESULTS

Task Assignment

Following the work by Stern (1999), in order to define a task for the Boolean network we start by arbitrarily selecting a subset of N_s nodes that we call *signal nodes*, $\{\sigma_{s_1}, \sigma_{s_2}, \ldots, \sigma_{s_{N_s}}\}$, from which we extract the *output signal* $R(t)$ defined as (see **Figure 1A**)

$$R(t) = \sum_{i=1}^{N_s} \sigma_{s_i}(t). \tag{1}$$

Assigning a task to the Boolean network consists in requiring that the output signal $R(t)$ approximates as much as possible a predefined *target function* (or *task*) $F(t)$ (see **Figure 1B**). In our model $F(t)$ is an arbitrary function such that $0 < F(t) < N_s$ for $1 \leq t \leq t_m$, where $t_m = 15$ is the number of time steps of the assigned task. We set $t_m = 15$ because this is the average number of time steps it takes for the network to stabilize its dynamics (see **Figure S1**). In biological terms, the task $F(t)$ would represent an expression pattern some genes must acquire in order for the organism to efficiently respond to a particular environmental challenge (like yeast responding to a heat shock). Since the networks are randomly constructed, it is expected that initially none of them have this response (their output signal $R(t)$ and the task $F(t)$ are usually different at the start of the simulation, see **Figure 1B**). Therefore, it is necessary to evolve the networks so that $R(t)$ approaches $F(t)$ as much as possible, as in **Figure 1C**. It is only through a series of mutations and adaptations that the phenotype $R(t)$ will approach $F(t)$ in some individuals, and then be transmitted to their offspring.

Throughout this work we use networks with $N = 50$ nodes, average connectivity $K = 2$ and $N_s = 12$ signal nodes (except in some figures where smaller networks are presented for illustrative purposes). The reason for this choice of parameters is the following. It has been observed that genetic networks of several real organisms are structured in functional modules, each one consisting of a few dozen genes or nodes

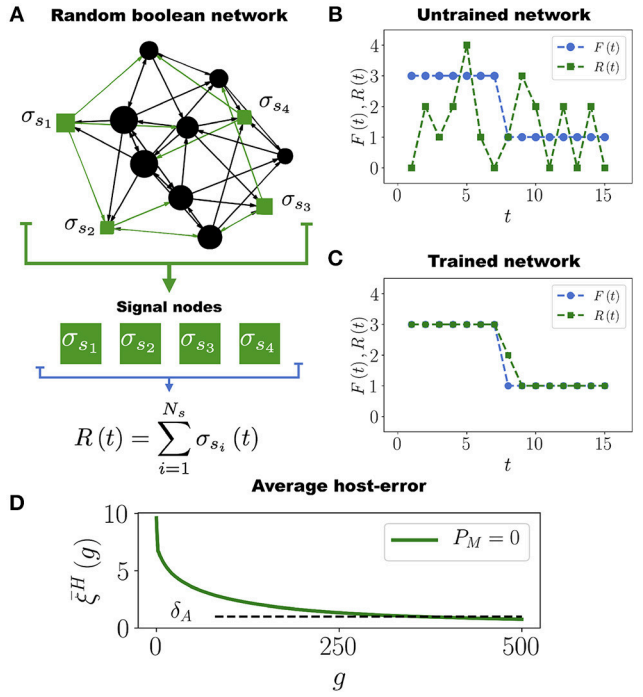


FIGURE 1 | Boolean Network Task. **(A)** Network with $N = 12$ nodes and $N_S = 4$ signal nodes (represented as green squares), which generate the output signal $R(t)$ that has to converge to the task $F(t)$. **(B)** Initially, the untrained network produces an output signal $R(t)$ (green curve with squares) very different from the target function $F(t)$ (blue curve with circles). **(C)** After the evolutionary process, the network is well adapted to perform the task. In this example, only one point of the output signal $R(t)$ acquires a wrong value in the time interval $1 \leq t \leq 15$ in which the task $F(t)$ is evaluated. **(D)** Average population error $\bar{\xi}^H$ as a function of the number of generations g, for a population with $P = 100$ networks, each having $N = 50$ nodes and $N_S = 12$ signal nodes. Note that $\bar{\xi}^H$ decreases and crosses the adaptation threshold $\delta_A = 1$ approximately at generation $g = 350$, after which most of the networks in the population become well adapted to the task $F(t)$.

(Resendis-Antonio et al., 2012). For instance, adaptive resistance to antibiotics in *Escherichia coli* is mediated by the MarA-AcrAB-TolC system which, when activated, produces efflux pumps that pump toxic molecules in the intracellular fluid out of the cell, keeping the internal antibiotic concentration below lethal levels. Activation of this system is controlled by a regulatory network consisting of about 15 nodes (Motta et al., 2015). Analogously, the cAMP-dependent protein kinase regulatory network (PKA-RN), which regulates (among other things) the stress response in *Saccharomyces cerevisiae*, consists of 15 nodes (Pérez-Landero et al., 2015). There are many more examples showing that specific cellular functions (such a the response to a given environmental challenge) are controlled by network modules composed of a few dozen nodes (Guo et al., 2016; Ma et al., 2017). Since in this work we are not considering any specific organism, we will assume in a generic way that the task $F(t)$ that the network has to acquire is encoded in $N_s = 12$ nodes, which in turn are embedded in a module of $N = 50$ nodes.

Finally, we perform all our simulations using Kauffman networks with connectivity $K = 2$ for two main reasons. First, these networks are trained faster than networks with connectivity smaller or larger than $K = 2$ (see **Figure S2**). A second, more fundamental reason is that networks with $K = 2$ exhibit critical dynamics, which means that their dynamical behavior is at the brink of a phase transition between order and chaos (Derrida and Pomeau, 1986; Aldana, 2003). Dynamical criticality confers the system interesting properties such as evolvability (i.e., the coexistence of robustness and adaptability) (Aldana et al., 2007; Torres-Sosa et al., 2012), faster information storage, processing and transfer (Langton, 1990; Nykter et al., 2008), and collective response to external stimuli without saturation (Kinouchi and Copelli, 2006), (or shorter training times, as in our case, see **Figure S2**). There is solid evidence indicating that gene regulatory networks of real organisms are dynamically critical or close to criticality (Shmulevich et al., 2005; Serra et al., 2007; Balleza et al., 2008; Daniels et al., 2018). Therefore, by choosing $K = 2$ we are working with a representative ensemble of networks that have an important dynamical property observed in real organisms.

Host Network Evolution

We consider a population of $P = 100$ networks, represented as $\{H_1, H_2, \ldots, H_P\}$, which have to perform the same task $F(t)$. We will refer to these networks as the *host networks* (HNs). At the start of the simulation all the HNs are identical replicas of one randomly constructed network. To make the output signal of the HNs approach the task $F(t)$ we implement a traditional evolutionary algorithm in which the networks are mutated, selected and replicated. Variability in the population is implemented by mutating the HNs with a mutation rate $\mu_H = 0.001$ per node per network per generation. Once a node σ_n of a given network H_i has been chosen for mutation, we perform any of the following changes with equal probability: (i) Randomly rewire one of the input or output connections of σ_n. (ii) Add a new input (or output) connection to σ_n from (or to) a randomly chosen node in the network. (iii) Remove one input or output connection of σ_n. (iv) Change one of the entries of the logical function f_n associated to σ_n.

The mutations described above can make each network H_i get closer to the task $F(t)$ or get away from it. To measure the adaptation of the HNs to the task we denote as $R_i(t)$ the output signal of the network H_i and define its adaptation error ξ_i^H as

$$\xi_i^H = (t_m)^{-1} \sum_{t=1}^{t_m} \left(R_i(t) - F(t)\right)^2 . \qquad (2)$$

Clearly, if $\xi_i^H = 0$ then the network H_i is perfectly trained (adapted) to perform the task $F(t)$, whereas large values of ξ_i^H indicate a poor adaptation. Therefore, when a mutation occurs such that ξ_i^H decreases, the adaptation of H_i increases and viceversa. We will say that the network H_i is *well adapted* to its task when $\xi_i^H \leq \delta_A$, where δ_A is the *adaptation threshold*. We set $\delta_A = 1$, which means that at most one node out of the N_s signal nodes is allowed to deviate one unit from the correct value at every time step during the interval $1 \leq t \leq t_m$ over which the task $F(t)$ is evaluated (see **Figure 1C**). The average population

error, defined as $\bar{\xi}^H = \frac{1}{P}\sum_{i=1}^{P}\xi_i^H$, measures the adaptation of the entire population to the task $F(t)$.

In each generation we mutate the HNs in the population with the mutation rate μ_H. Then, we choose the 10 best networks (those whose errors ξ_i^H have the lowest values) to get through the next generation while the other 90 networks are removed from the simulation. These 10 networks are replicated by making 9 copies of each one in order to restore the population to its original size $P = 100$. This evolutionary process is repeated until the population crosses the adaptation threshold. A "generation" consists in a full round of mutation, selection and replication processes. **Figure 1D** shows that the average population error $\bar{\xi}^H$ decreases throughout generations. This is expected since at each generation we select the networks that minimize the error. From **Figure 1D** we see that it takes about 350 generations for the average population error of hosts networks to cross δ_A and become well adapted to the task (see also **Movie S1**). We have performed simulations with smaller values of the adaptation threshold: $\delta_A = 0.5$ and $\delta_A = 0.2$, and the results are qualitatively the same. The only difference is that the smaller the value of δ_A, the longer the computing time for the average population error $\bar{\xi}^H$ to cross this threshold (see **Figure S3**). The results presented in **Figure 1D** correspond to a population of HNs evolving by themselves, i.e., without interacting among them or with other networks. We refer to this case as the *control case*.

Interaction With the Microbiota: Holobiont Evolution

To model the interaction between the host organism and its microbiota we allow the training of each host network H to be assisted by a set of P_M other networks, $\mathcal{B} = \{M_1, M_2, \ldots, M_{P_M}\}$, each one representing a *microbial network* (MN). We will refer to the set $\mathcal{B}$ as the microbiota, and to the set $\mathcal{L} = \{H, \mathcal{B}\} = \{H, M_1, \ldots, M_{P_M}\}$, as the holobiont.

Each microbial network $M_j \in \mathcal{B}$ also has to perform a predefined task $F_j^M(t)$, which is an arbitrary function constructed in the same way as the host-network task $F(t)$. The microbial tasks $F_1^M(t), \ldots, F_{P_M}^M(t)$ are different from each other and from $F(t)$. Before the training of H begins, each $M_j \in \mathcal{B}$ is previously trained to be well adapted to its own task $F_j^M(t)$. This means that all the microbial errors ξ_j^M satisfy, from the very beginning, the well-adapted condition $\xi_j^M < \delta_A$ (as in **Figure 1D**; the microbial error ξ_j^M is defined similarly as in Equation (2); see the M&M section for the precise definition). Thus, at generation $g = 0$ the holobiont consists of the untrained host network H and a set of well adapted MNs. The evolution of the holobiont then proceeds with the adaptation of H to its task and allowing it to interact (as described below) with MNs that already have their own interests. The rationale behind this initial setup is twofold. First, allowing the training of H to be assisted by well-adapted MNs captures the fact that at any moment during its evolution, the host organism can recruit from the environment microbial populations already adapted to their environments and able to carry out some functions by their own. Second, we want to determine whether evolutionary conflict emerges between the host and microbial networks when the holobiont evolves as a unit of adaptation, as has been pointed out in Moran and Sloan (2015) and Douglas and Werren (2016). Such a conflict would be apparent in our simulations if a reduction of the host-network error ξ^H occurs with a simultaneous increase in the average microbial error $\bar{\xi}^M$, or viceversa.

The interaction between H and its microbiota $\mathcal{B}$ is implemented as follows (see **Figures 2A,B**). Consider the case where a given node σ_n of H has been chosen for mutation such that a new input (output) connection is to be added. Then this new connection can be selected with equal probability either within H itself or from any of the microbial networks $M_j \in \mathcal{B}$. Likewise, when a given node of a microbial network M_j is mutated so as to receive a new connection (either input or output), the new connection can be established within M_j itself, with H or with any other microbial network $M_k \in \mathcal{B}$. This allows the emergence of regulatory interactions between all the networks that constitute the holobiont.

For the adaptation of H to its task we consider the evolution of a population of $P = 100$ holobionts. Throughout the evolutionary process all the networks in each holobiont undergo the same kind of random mutations described in the section Host Network Evolution (with the possibility of interactions across networks, as mentioned in the previous paragraph). However, in our simulations the mutation rate μ_M for the MNs is ten times larger than the mutation rate μ_H for the host network, namely $\mu_M = 10\mu_H$. This captures the fact that bacterial colonies, due to their high reproduction rates, develop mutants at least ten times faster than populations of eukaryotic cells in multicellular organisms (Lynch, 2010; Lynch et al., 2016). It is important to emphasize that in our model each network in the holobiont has to be considered not as representing a single cell, but an entire cell population. In each generation, holobionts are ranked according to their error ξ^L, and the ten with the smallest errors are selected for reproduction (see M&M). In all further simulations the unit of selection is the holobiont, as in each generation we select the ten best holobionts (based on the error ξ^L that takes into account the host and microbial errors) and replicate them.

Figure 2C shows the average population error $\bar{\xi}^H$ of the host network H across generations for holobionts as well as for the control case (host networks evolving by themselves without interacting with microbial networks). In the simulations reported in **Figure 2C** each holobiont consists of one host network H and one microbial network M ($P_M = 1$). It is clear that interacting with only one microbial network M already makes H to adapt much faster to its task than evolving on its own. In the holobiont case, the error $\bar{\xi}^H$ crosses the adaptability threshold δ_A in about one fourth of the generations required for the control case to do it (see **Movie S2** and compare with **Movie S1**). Furthermore, the final error after 500 generations is considerably smaller for the holobiont case ($\bar{\xi}^H \approx 0.2$) than for the control case ($\bar{\xi}^H \approx 0.95$). Note that an error $\bar{\xi}^H \approx 0.2$ means that, on average, at most 3 points of the output signal $R(t)$ deviate one unit from the task $F(t)$ in the whole interval $1 \leq t \leq 15$, which represents a percent error $100 \times 3/(N_s \times 15) \approx 1.6\%$. This is almost a perfect adaptation hard to achieve in the control case. For the control case, it takes about 3000 generations to reach a similar error of $\bar{\xi}^H = 0.2$ (See **Figure S3**). The average microbial error $\bar{\xi}^M$ also decreases,

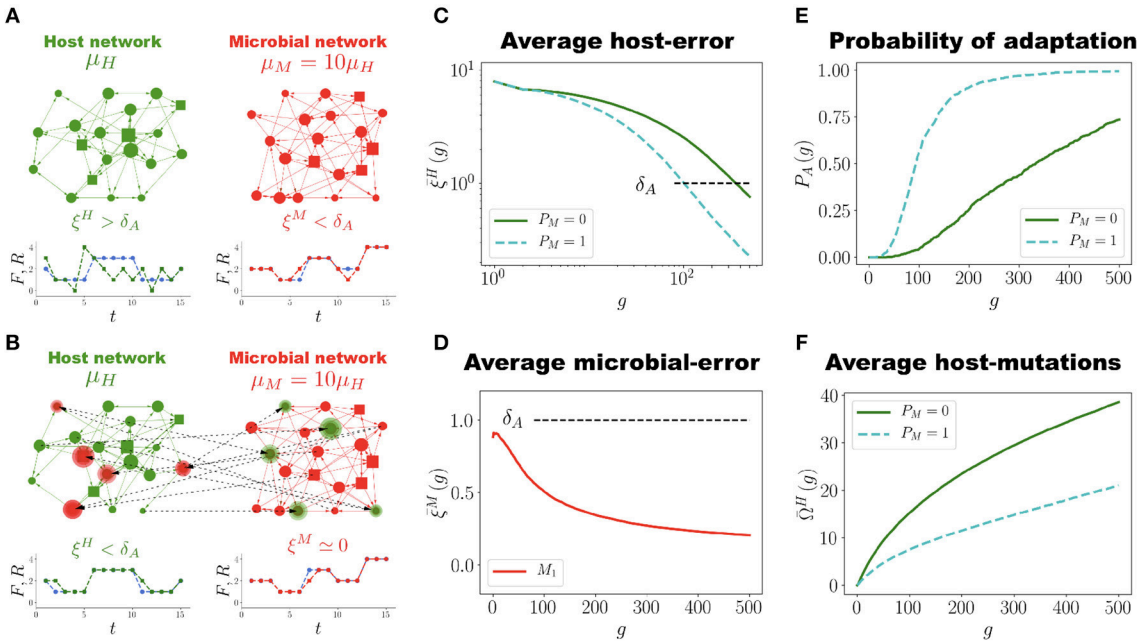


FIGURE 2 | Network Coevolution. **(A)**. Schematic representation of the holobiont, which in this case consists of one host network H (green) and one microbial network M (red), each with $N = 20$ nodes and $N_S = 4$ signal nodes (represented by squares). At generation $g = 0$ the host network H is not adapted to its task ($\xi^H > \delta_A$), whereas the microbial network M is well adapted ($\xi^M < \delta_A$). The mutation rates μ_H and μ_M of the host and microbial networks, respectively, satisfy $\mu_M = 10\mu_H$. **(B)** At generation $g = 150$ regulatory interactions between H and M have been established (dashed lines). The highlighted nodes in each network have regulators in the other network. H has become well adapted to its task ($\xi^H < \delta_A$) while the microbial error ξ^M has decreased almost to zero. **(C)** Population average host error $\bar{\xi}^H$ as a function of generations for the holobiont case (H and M evolving together, blue dashed curve, $P_M = 1$) and for the control case (H evolving by itself, green solid curve, $P_M = 0$). In the holobiont case the adaptability threshold δ_A is reached faster ($g \approx 100$) than in the control case ($g \approx 350$). Also, at the end of the simulation ($g = 500$) the error for the holobiont case is about five times smaller than for the control case. **(D)** Evolution of the average microbial error $\bar{\xi}^M$. At generation $g = 0$, $\bar{\xi}^M$ already satisfies the well-adapted condition, $\bar{\xi}^M < \delta_A$, but it further decreases as the evolution of the holobiont goes on. **(E)** Probability of adaptation $P_A(g)$ across generations for the holobiont (blue dashed curve) and control (green solid curve) cases. Note that $P_A(g)$ increases and saturates faster in the holobiont case. **(F)** Average number $\bar{\Omega}^H(g)$ of accumulated mutations in the host network H during its adaptation process for the holobiont case (blue dashed curve) and the control case (green solid curve). Interacting with the microbial network halves the number of mutations H has to undergo in order to adapt to its task. The numerical simulations to generate the graphs (C) to **(F)** were carried out using networks with $N = 50$, $N_S = 12$ and populations of $P = 100$ networks.

as shown in **Figure 2D**. At generation $g = 0$, $\bar{\xi}^M$ is already below the adaptability threshold δ_A, but it decreases even further as the evolution of holobionts proceeds. Therefore, the adaptation of the holobiont takes place with no conflict of interest between H and its microbial networks.

In **Figure 2E** we report the probability of adaptation $P_A(g)$, defined as the fraction of holobionts in which the host-network error ξ^H crosses the adaptation threshold δ_A at generation g. It is apparent from **Figure 2E** that this probability for the holobiont case increases and saturates much faster than for the control case. About 80% of the holobionts are well adapted after only 120 generations, whereas host networks evolving by themselves never reach 75% of adaptation during the whole simulation time. In addition to speeding up and increasing the adaptation of H, the interaction between H and M also considerably reduces the number of mutations H has to accumulate in order to adapt to its task, as **Figure 2F** reveals. This is not a trivial result, for only the mutations in both H and M that increase the adaptation of the holobiont are selected and fixed in the population. Thus, even though M mutates ten times faster than H, not all of those mutations are beneficial to the adaptation of the holobiont and consequently, not all of them become fixed in the population. Actually, from **Figure 2E** we observe that the average number $\bar{\Omega}^H$ of accumulated mutations in H to reach the adaptation threshold δ_A is not ten, but only two times larger for the control case than for the holobiont case. However, it is true that because μ_M is larger than μ_H the adaptation of H is improved (our simulations show that there is no significant difference between the holobiont and control cases when $\mu_M = \mu_H$, see **Figure S4**).

Symbiosis and Dysbiosis

To show that symbiotic relationships emerge between the host and microbial networks, once the holobiont is well adapted (after 500 generations as in **Figure 2C,D**), we remove the connections between H and M (the dashed lines in **Figure 2B**) and compute the errors of each network at performing their respective tasks while disconnected. This can be thought of as an antibiotic administration where several bacterial species are removed from the microbial population, or as trying to cultivate these symbiotic microbes without their respective host. Thus, a set of microbial species, represented by M, are removed from the holobiont and then the fitness of the host is evaluated without them. At the same

time, we determine the survivability of these microbial species M in the absence of their host. Since M starts the evolutionary process already well adapted to its task, one can expect that its error does not significantly increase after the connections between H and M are removed. However, **Figure 3** shows a typical example in which removing the connections between H and M increases both errors ξ^H and ξ^M to values that correspond to untrained networks. Thus, in the example shown in **Figure 3**, after the holobiont has been adapted as a whole, none of the networks it consists of can perform their respective tasks when separated (see **Figure S5** for population statistical averages).

Multitasking and Microbial Diversity

So far we have presented results in which the host network H has to perform only one task. Interaction with one MN significantly improves the adaptation of H (and of the holobiont) and reduces the number of mutations it has to undergo in order to become well adapted. It could be expected that adding more MNs to the microbiota would further enhance the adaptation of H. However, this is not the case. Adding more MNs either has no effect or can even worsen the adaptation of the host (see **Movie S3** and **Figure S6**). This result is in contradiction with the great diversity observed in the microbiota of real organisms and the ability of the holobiont to adequately perform multiple tasks.

For this reason, we now consider the case in which the host network H is trained to perform T multiple tasks $F_1(t), F_2(t), \ldots, F_T(t)$, each being an arbitrary function constructed as described in the Task Assignment section. Since Boolean Network dynamics are deterministic, depending on the initial condition the dynamics of H will be set to follow a specific task $F_\tau(t)$. We can measure the adaptation errors ξ_τ^H and $\xi_{j,\tau}^M$ of H and the microbial network $M_j \in \mathcal{B}$, respectively, when H is being trained to perform the particular task $F_\tau(t)$ (see the M&M section for a precise definition of ξ_τ^H and $\xi_{j,\tau}^M$). This allows us to compute the adaptation of the holobiont separately for each task. Averaging ξ_τ^H and $\xi_{j,\tau}^M$ over all the tasks gives us the total adaptation errors ξ^H and ξ_j^M for the host and microbial networks, respectively (see the M&M section).

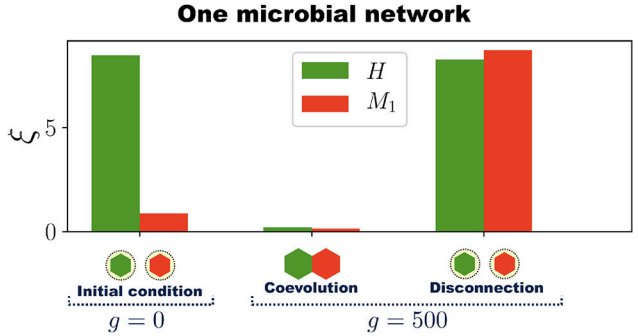


FIGURE 3 | Emergence of symbiosis and dysbiosis. A holobiont consisting of one host network H and one microbial network M_1 evolves for 500 generations. Then H and M_1 are disconnected and the errors at performing their respective tasks evaluated (H and M_1 are represented by hexagons at the bottom of the bar chart). At generation $g = 0$, when the training of H begins, the host-network error ξ^H (green bar) is large whereas the microbial-network error ξ^M (red bar) is already below the adaptation threshold δ_A. After H and M_1 have coevolved for 500 generations both ξ^H and ξ^M are quite below δ_A, which indicates the adaptation of the entire holobiont. Then, H and M_1 are disconnected and their respective errors evaluated. Note that after disconnection both errors ξ^H and ξ^M in this example increase to levels corresponding to completely untrained networks. The simulations were performed with networks having $N = 50$ nodes and $N_S = 12$ signal nodes.

We implement two ways in which the MNs can assist the adaptation of H to perform many different tasks. First, there is the *non-specialized interaction* case in which all the MNs can interact among each other and with H. Also, all the MNs can participate in the adaptation of H to all of its tasks (see **Figure 4A**). In each generation the networks are mutated, allowing new interactions to appear between any two networks within the holobiont. This means that new incoming or outgoing connections can be established either between H and any of its MNs, or between any two MNs. We consider again a population of $P = 100$ holobionts. After the networks in each holobiont have been mutated (with the mutation rates μ_H and μ_M for the host and microbial networks, respectively), the ten best holobionts are selected and replicated (see the M&M section for a definition of the holobiont error ξ^L in the multitasking case). **Figure 4B** shows the population average $\bar{\xi}^H$ of the host-network error for the case in which H has to perform 10 different tasks. It is clear from this figure that adding more than one microbial network to the microbiota has no effect on the adaptation of H to its 10 different tasks. Therefore, in the non-specialized case increasing the diversity of the microbiota does not help the adaptation rate of the host.

As a second alternative we implement a *specialized interaction* in the microbial networks. In this case the microbiota $\mathcal{B} = \{M_1, \ldots, M_{P_M}\}$ is divided into P_G disjoint non-empty subsets, or "niches", $\{G_1, G_2, \ldots, G_{P_G}\}$. The set of tasks $\mathcal{F} = \{F_1, \ldots, F_T\}$ is also partitioned, as evenly as possible, into P_G non-overlapping subsets, $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_{P_G}\}$. The maximum number of niches is $P_G = T$, for in this case each subset $\mathcal{T}_\tau$ contains only one task. To each niche G_τ we associate a subset $\mathcal{T}_\tau$ of tasks (see **Figure 4C**). The host network H is still trained to perform the T different tasks $F_1, \ldots, F_T$. However, the training of H to the tasks in the particular set $\mathcal{T}_\tau$ is assisted only by the networks in the corresponding niche G_τ. For each niche G_τ we compute an error ξ_τ^G that measures the adaptation of the holobiont when H is being trained to perform the tasks in the specific subset $\mathcal{T}_\tau$ (see the M&M section for a precise definition of the niche error ξ_τ^G). During the adaptation of H to the tasks in $\mathcal{T}_\tau$, only the mutations in H or in the microbial networks belonging to G_τ that reduce the corresponding error ξ_τ^G are selected. The important point to note here is that the adaptation of H to its tasks can be measured separately for each niche. The holobiont error ξ^L is computed as the average of the errors ξ_τ^G over all the niches (see the M&M section).

During the training of H, the MNs in one niche can develop interactions between them, but they cannot interact with the MNs in a different niche, as **Figure 4C** indicates. This is consistent with the observation that multicellular organisms maintain the stability of its microbiota by reducing microbial interactions (Deines et al., 2017). If microbes interacted with no

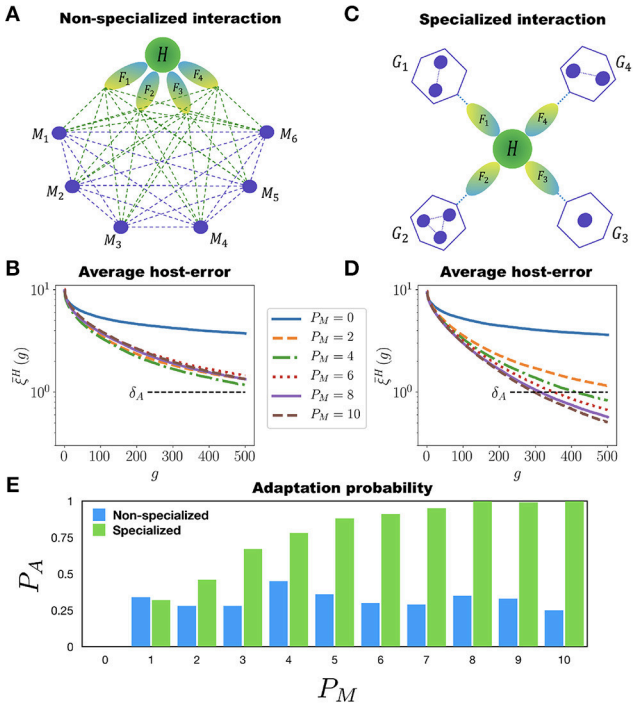


FIGURE 4 | Multitasking and microbial specialization. The host network H (green circle) has to perform T different tasks $F_1, F_2, \ldots, F_T$ (represented as ellipses) assisted by P_M microbial networks $M_1, M_2, \ldots, M_{P_M}$ (small circles). **(A)** Schematic representation of a holobiont with non-specialized interaction in the microbial networks. Each microbial network can interact with all the other ones and participate in the adaptation of H to all of its tasks. The dashed lines represent possible interactions. **(B)** Evolution of the host-network error $\bar{\xi}^H$ in the non-specialized case for $T = 10$. Each curve corresponds to a different value of P_M, starting from $P_M = 0$ (H evolving by itself) up to $P_M = 10$. In none of these cases $\bar{\xi}^H$ crosses the adaptation threshold δ_A even after 500 generations. Furthermore, adding more than one microbial network to the holobiont has a negligible effect on the adaptation of H. **(C)** Schematic representation of the holobiont with specialized interaction in the microbial networks. The set of P_M microbial networks is divided into T different niches, $G_1, \ldots, G_T$, each one participating in the adaptation of H to only one task. The microbial networks can interact among them only if they belong to the same niche. **(D)** Host-network error $\bar{\xi}^H$ across generations for the specialized interaction and $T = 10$. Again, each curve corresponds to a different number P_M of microbial networks. Note that in this case the more microbial networks in the holobiont the better the adaptation of H. Note also that for $P_M = 10$ the host-network error $\bar{\xi}^H$ crosses the adaptation threshold δ_A in less than 300 generations. **(E)** Adaptation probability P_A at generation $g = 500$ as a function of the number P_M of microbial networks participating in the host's adaptation. The blue and green bars correspond to the non-specialized and specialized cases, respectively. Note that in the non-specialized case the adaptation probability remains low regardless of the number of microbial networks, whereas in the specialized case, the more microbial networks the better the adaptation of the host. The simulations were carried out for populations of $P = 100$ holobionts and networks with $N = 50$ nodes.

organization, the loss of one microbial species would affect the fitness of all the others, increasing the risk of extinction cascades (Coyte et al., 2015). Therefore, in the specialized interaction case we compartmentalize the MNs allowing interactions among them only if they belong to the same niche (all the MNs in all the niches can, of course, interact with the host network H).

Figure 4D shows the evolution of the average host-error $\bar{\xi}^H$ for simple case where each niche has one MN ($P_G \leq T$). It is clear from **Figure 4D** that, contrary to the non-specialized case, adding more MNs to the microbiota in a specialized way considerably improves the adaptation of H to its multiple tasks. Furthermore, in **Figure 4E** we report the probability of adaptation $P_A(500)$ at generation $g = 500$ as a function of P_M for both the non-specialized and specialized cases. In the former case P_A remains low and never improves as P_M increases, whereas in the later case P_A monotonously grows with P_M. This clearly shows that both diversity and specialization of the MNs are necessary for H to adapt to multiple tasks.

The results presented for the specialized interaction case also hold when every niche is populated with more than one MN ($P_M \geq 2T$). In **Figure 5A** (see also **Movie S4**) we report the evolution of a holobiont with $T = 10$ different tasks and the same number of niches, and $P_M = 25$ microbial networks (each niche contains either two or three MNs). At generation $g = 0$, H is poorly adapted to all of its tasks (represented in red), whereas all the MNs are already well adapted (represented in blue). As the evolution proceeds H becomes more adapted to all of its tasks. Furthermore, the microbial networks also become more adapted to their own tasks. Note that the adaptation of H to its different tasks occurs at different rates, as can be seen from the color-code of the tasks through the holobiont evolution. This is consistent with the observation that different symbiotic relationships between the host and its microbial communities emerge at different rates (Doolittle and Booth, 2017). In the example shown in **Figure 5A** the adaptation of the whole holobiont crosses the adaptation threshold δ_A in <300 generations. Interestingly, the same results are obtained in the specialized case when many more microbial networks are introduced into the holobiont (see **Figure S7**).

The specialized interaction scheme allows us to compute the robustness of the holobiont under loss of microbial diversity. For this, once the holobiont is well adapted, we disconnect Δn microbial networks from it and compute the resulting host-network error ξ^H averaged over all the host's tasks. **Figure 5B** shows that ξ^H gradually increases as more microbial networks are disconnected from the holobiont. Therefore, a loss in microbial diversity clearly reduces the adaptation of the host.

DISCUSSION

Multicellular organisms and microbes have coevolved in many different ways, not only as holobionts being units of adaptation (Theis et al., 2016). However, the persistence across generations of regulatory interactions between the host and its microbes is a necessary condition for natural selection to operate at the holobiont level (Doolittle and Booth, 2017). These regulatory interactions have to preserve the holobiont's functionality regardless of the specific microbial species that generate them. This is the "It's the song not the singer" (ITSNTS) approach to evolution proposed by Doolittle (Doolittle and Booth, 2017; Doolittle and Inkpen, 2018) and exemplified by Taxis et al. in ruminal ecosystems (Taxis et al., 2015). In this work we have

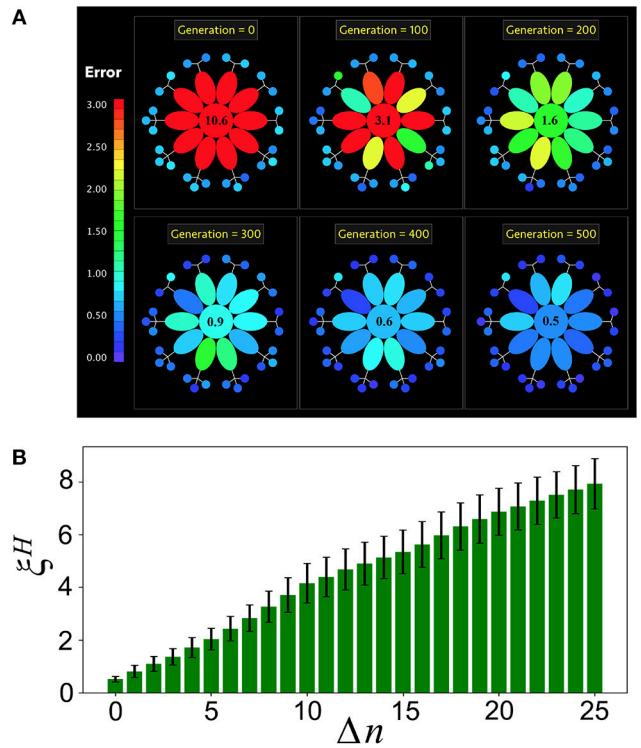


FIGURE 5 | Holobiont's complexity and loss of microbial diversity. **(A)** Snapshots of the evolution of a holobiont in the specialized case. The holobiont consists of $T = 10$ tasks (ellipses) and $P_M = 25$ microbial networks (small circles). All errors larger than 3 are colored in red. At generation $g = 0$ the host network H has a large error in all of its tasks, while the bacterial networks are well adapted. As the evolution proceeds, the error of the entire holobiont decreases, but the adaptation of H to its different tasks takes place at different rates. The number at the center is the value of the host error $\bar{\xi}^H$ averaged over all of its tasks. **(B)** After the holobiont is well adapted at generation $g = 500$, Δn microbial networks are disconnected and the resulting host-network error $\bar{\xi}^H$ is computed. This simulates the effect on the holobiont's adaptation of a reduced microbial diversity. Note that $\bar{\xi}^H$ gradually increases as more microbial networks are disconnected, indicating that the host becomes less adapted as the microbial diversity decreases.

incorporated into a single evolutionary model both the concept that the holobiont is a unit of selection and Doolittle's ITSNTS approach. We have done so by requiring in our simulations that only the best adapted holobionts at each generation are the ones able to go throughout the selective filter, pass to the next generation and replicate all of its constituent networks. However, in our model selection acts on a dynamical property of the holobiont, which is the host's output signal, in order to bring it close to the functions the host needs to perform. In this scheme, it does not matter what nodes or microbial networks participate in the regulation of the dynamical functions. What is important is the preservation across generations of the dynamical functions themselves, and this must hold for both the host and the microbial networks (which also have to perform and preserve their own functions). Thus, although the holobiont might not be the only unit of selection (Theis et al., 2016), we have concentrated on those important host-microbe co-interactions that transmit functionality across multiple generations (Roughgarden et al., 2018). Our model does not assume that the microbial networks reside inside the host, but only that they interact with it and that the host-microbe interactions are transmitted across generations. This propagation can happen in various ways other than vertical transmission from parents to offspring as, for instance, when the host constructs its environment with a stable microbial composition (Fitzpatrick, 2014).

We have shown that the host network can actually be trained to perform one task without the help of any microbial networks, as **Figure 1D** and **Figure S3** illustrate. However, allowing interactions between the host and microbial networks greatly speeds up and improves the adaptation of the entire holobiont. This is because the host network does not only adapt to its tasks faster, better and with less mutations when it is allowed to interact with microbial networks, but the microbial networks themselves considerably improve their own adaptation to their respective tasks. Furthermore, adaptation of the host network to perform *multiple* tasks is improved only when it is allowed to interact with a diverse and specialized microbiota, as **Figure 4D** shows.

In light of these results, we observe that the microbiota does not only help the host to adapt to its tasks. There is mutual benefit in which both the host and its microbial communities contribute to each other's adaptation. It is in this sense that the holobiont can be considered as an evolutionary unit.

It is important to mention that in our model the holobiont cannot just be considered as one "big network" evolving to perform a set of tasks. There are two essential aspects that have to be emphasized. First, the rate at which mutants are generated μ_M in the microbial networks is considerably larger than that μ_H of the host networks. Second, the set of microbial networks must be partitioned into disjoint (i.e., non-interacting) niches for the host network to efficiently adapt to multiple tasks, where each niche specializes in the adaptation of the host to one specific subset of tasks. These two aspects provide the holobiont with a complex internal dynamical structure that prevents us from viewing it as just one big network (see **Figure 5A** and **Figure S7**). A holobiont for which $\mu_M = \mu_H$ and the microbial networks are not partitioned into specialized niches, could be considered as a single large homogeneous network. But in such a homogeneous case the holobiont evolution benefits neither from the host-microbe interactions nor from the microbiota's diversity (see **Figures S4**, **S8**). Rather, the structural and dynamical internal complexity of the holobiont, embodied in the functional modularity and specialization of the microbial niches as well as in the difference between the host and microbial mutation rates, is required to facilitate and improve the holobiont's adaptation to perform multiple tasks. Hence, our results show that complexity, modularity and functional specialization are necessary properties that naturally facilitate the evolution and adaptation of the holobiont as well as the diversification of the microbiota (Sachs et al., 2014), whereas structural and functional homogeneity is either inconsequential or even detrimental to the holobiont's evolution.

One may wonder whether the relationship $\mu_M = 10\mu_H$ between the microbial and host network's mutant-generating

rates accurately reflects reality. We have explored a wide range of values of the ratio $\gamma = \mu_M/\mu_H$, ranging from $\gamma = 1$ to $\gamma = \infty$. The latter case corresponds to $\mu_H = 0$, which means that the adaptation of the host to its task does not occur across generations, but within the host's lifespan. In this extreme case, the adaptation of the host to its task occurs due to mutations in the microbiota but not in the host itself. Our simulations show that the adaptation of the host network is almost equally accelerated and improved for $\gamma = 10$ than for $\gamma = \infty$ (see **Figure S9**).

In our model the host network interacts with microbial networks which, from the very beginning, are already well adapted to perform their own functions. The reason for this is to determine whether or not the well-adapted condition imposed on the microbial networks represents a restriction that could generate evolutionary conflict within the holobiont. It has been pointed out that the emergence of symbiotic relationships between organisms requires the symbionts to be highly cooperative and show very little conflict (Morris et al., 2012; Sachs and Hollowell, 2012; Sachs et al., 2014; Queller and Strassmann, 2016). Our simulations show that the evolution of the holobiont can very well take place with no evolutionary conflict between its constituent networks, as long as the microbiota is partitioned into specialized niches. This modularization and division of labor are essential to prevent microbial competition and within-group conflict in the holobiont (West et al., 2015) (see **Figure S6**). Additionally, modularization of the microbiota allows the holobiont to acquire new functions without affecting the ones already present.

Interestingly, similar results regarding the adaptation of the host network to its tasks are obtained when the well-adapted condition is not imposed on the microbial networks. Our simulations show that the adaptation of the host network is equally accelerated and improved when it interacts with microbial networks that do not have to perform any task (see **Movie S5**). However, even when the microbial networks are free of any selective pressure, their dynamics are stabilized when they coevolve with the host network (see **Movie S5**). This is important because it can be interpreted as the holobiont acquiring, at any moment, microbes from the environment and coevolving with them, generating intergenomic epistasis that reduces within-group conflict and promotes the adaptation of the entire holobiont (Bordenstein and Theis, 2015).

Finally, we would like to mention that, although there exist many qualitative taxonomical studies showing the existence of a great variety of host-microbe symbiotic interactions, there are very few mathematical and computational models aiming to explain the general mechanisms responsible for the emergence of such interactions and the need for diversification and specialization of the microbiota (Manor et al., 2014). We have not explicitly considered competition or parasitism in our model. However, by integrating the ITSNTS approach with the hologenome hypothesis (the holobiont as a unit of selection Rosenberg and Zilber-Rosenberg, 2016; Roughgarden et al., 2018), we were able to reproduce many of the observed behaviors in the evolution of holobionts, such as reduction of evolutionary conflict, division of labor, emergence of symbiotic interactions and dysbiosis when the microbiota diversity is reduced. Our model may thus lay the foundations for a comprehensive understanding of the long-lasting coevolution of multicellular organisms and microbes.

MATERIALS AND METHODS

Boolean Network Model

The Boolean network consists of a set of N nodes $\{\sigma_1, \sigma_2, \cdots, \sigma_N\}$, each acquiring the values 0 or 1 that represent two possible states of activity: "active" or "inactive." The value of each node σ_n is determined through a logical function f_n that depends on a set of k_n other nodes in the network denoted as $I_n = \left\{\sigma_1^n, \sigma_2^n, \cdots, \sigma_{k_n}^n\right\}$. The nodes in the set I_n are known as the *inputs* or *regulators* of σ_n. In the context of genetic networks these regulators together with the logical function f_n mimic the effect of k_n transcription factors (synthesized by the regulators) acting on the expression of σ_n. For networks of real organisms both the logical function f_n and the set of regulators I_n associated to each gene are carefully constructed according to the activating and inhibitory nature of the regulatory interactions between the genes. Here, the k_n regulators of each node σ_n are randomly chosen from anywhere in the network. The logical functions f_n are also randomly chosen from the ensemble of all possible logical functions with k_n variables. In this work we start the simulations with networks for which $k_n = K = 2$, which means that every node in the initial networks has exactly $K = 2$ regulators (chosen randomly). Note that this is a directed network because if node σ_m is a regulator of σ_n, it does not necessarily happen that σ_n is also a regulator of σ_m (although it may happen). Note also that throughout the evolution of the networks some input and output connections are added to, or removed from, different nodes. Therefore, the final networks do not have a constant connectivity $K = 2$ for every node. The final networks will have a connectivity distribution similar to the one observed in the Erdös-Rényi topology with an average around $K = 2$ (see **Movies S1**, **S2**). Once each node in the network has been provided with a set of inputs and a logical function, the network dynamics is given by

$$\sigma_n(t+1) = f_n\left(\sigma_1^n(t), \sigma_2^n(t), \cdots, \sigma_{k_n}^n(t)\right). \tag{3}$$

Starting the dynamics from an initial condition $\Sigma = \left\{\sigma_1(0), \sigma_2(0), \cdots, \sigma_N(0)\right\}$, the network transits throughout a series of states until a periodic pattern is reached, which is known as a *dynamical attractor*. There is a great body of work showing that the dynamical attractors of the network correspond to different cell types or cell fates (or more generally, to different functional states of the cell). Here we are not interested in the dynamical attractors, but in training the network to perform a specific task.

Microbial and Holobiont Errors: One Task

Let us consider a holobiont $\mathcal{L} = \left\{H, M_1, \ldots, M_{P_M}\right\}$. When the host network H has to perform only one task $F(t)$, its error ξ^H is given in Equation (2). We similarly define the microbial error ξ_j^M corresponding to the j^{th} microbial network M_j as $\xi_j^M = \frac{1}{t_m}\sum_{t=1}^{t_m}\left(R_j^M(t) - F_j^M(t)\right)^2$, where $R_j^M(t)$ and $F_j^M(t)$ are

the output signal and target function of M_j, respectively. Different tasks are assigned to the different microbial networks. The error ξ^L of the entire holobiont is then computed as

$$\xi^L = \frac{1}{1+P_M}\left(\xi^H + \sum_{j=1}^{P_M}\xi_j^M\right). \quad (4)$$

In our simulations the population contains $P = 100$ holobionts, $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_P$. For each holobiont $\mathcal{L}_i$ we compute its error ξ_i^L as in Equation (4), which is then used at each generation to select the best holobionts in the population.

We also preformed numerical simulations using the following definition for the holobiont error:

$$\xi^L = \frac{1}{2}\left(\xi^H + \frac{1}{P_M}\sum_{j=1}^{P_M}\xi_j^M\right). \quad (5)$$

The difference between Equations (4 and 5) is the contribution of the microbiota to the holobiont's error. In Equation (4) the contribution of the total microbial error could be very large as compared to the contribution of the host-network error, especially if there are many microbial networks in the holobiont. Contrary to this, in Equation (5) the microbiota and the host have the same contribution regardless of the number of microbial networks in the microbiota. Our simulations show that both definitions produce qualitatively the same results (see **Figure S6B**). This is because at each generation in the evolutionary process we are selecting the best holobionts in the population, and selecting the best holobionts eventually leads to the same type of individuals regardless of the way in which the contribution of each particular network is weighted. In this work we present results using the definition given in Equation (5).

Host and Microbial Errors: Multitasking Non-specialized Case

Let us consider a holobiont $\mathcal{L} = \{H, M_1, \ldots, M_{P_M}\}$ where now the host network H has to perform T different tasks $F_1(t), F_2(t), \ldots, F_T(t)$. For each task $F_\tau(t)$ the network starts its dynamics from a predefined initial condition $\Sigma_\tau = \{\sigma_1^\tau(0), \sigma_2^\tau(0), \ldots, \sigma_N^\tau(0)\}$. Let us denote as $R_\tau(t)$ the output signal of H when it starts its dynamics from the initial condition Σ_τ. The error ξ_τ^H corresponding to the task $F_\tau(t)$ is computed as

$$\xi_\tau^H = \frac{1}{t_m}\sum_{t=1}^{t_m}\left(R_\tau(t) - F_\tau(t)\right)^2. \quad (6)$$

This allows us to measure the adaptation of the host network to each of its tasks separately. The total adaptation error ξ^H of H is computed by averaging the corresponding errors over all the T tasks that H has to perform: $\xi^H = T^{-1}\sum_{\tau=1}^{T}\xi_\tau^H$.

To define the error $\xi_{j,\tau}^M$ of the microbial network M_j when the host network is being trained to perform the task $F_\tau(t)$ we have to consider the fact that H may be regulating some of the nodes of M_j (throughout the evolution of the holobiont such regulations may appear). Therefore, the output signal of M_j depends on the initial condition Σ_τ used to start the dynamics of H. Let us denote as $R_{j,\tau}^M$ the output signal of M_j when H started its dynamics from the initial condition Σ_τ. The corresponding microbial error $\xi_{j,\tau}^M$ is then defined as

$$\xi_{j,\tau}^M = \frac{1}{t_m}\sum_{t=1}^{t_m}\left(R_{j,\tau}^M(t) - F_j^M(t)\right)^2, \quad (7)$$

where F_j^M is the task assigned to the microbial network M_j (this network was already well adapted to its tasks and has to remain so during the evolutionary process). The total microbial error ξ_j^M corresponding to M_j is then computed by averaging $\xi_{j,\tau}^M$ over all the tasks: $\xi_j^M = T^{-1}\sum_{\tau=1}^{T}\xi_{j,\tau}^M$. The holobiont error ξ^L is computed using Equation (4), where now ξ^H and ξ_j^M are the host and microbial errors averaged over all the tasks as described above.

Niche Error: Multitasking Specialized Case

Let us consider the niche $G_\tau = \{M_{\tau_1}, M_{\tau_2}, \ldots, M_{\tau_{p_\tau}}\}$, containing p_τ microbial networks. This niche is helping H to adapt to the tasks in the set $\mathcal{T}_\tau = \{F_{\tau_1}, F_{\tau_2}, \ldots, F_{\tau_{q_\tau}}\}$, which contains q_τ tasks. The error ξ_τ^G corresponding to this niche is defined as

$$\xi_\tau^G = \frac{1}{1+p_\tau}\left(\xi_\tau^H + \sum_{i=1}^{p_\tau}\frac{1}{q_\tau}\sum_{j=1}^{q_\tau}\xi_{\tau_i,\tau_j}^M\right), \quad (8)$$

where ξ_τ^H and ξ_{τ_i,τ_j}^M are defined in Equations (6, 7), respectively (in the latter case the subscripts j and τ change to τ_i and τ_j respectively, since we have to account for the different microbial networks and functions associated to the niche G_τ).

The holobiont error for the specialized interaction case is computed by averaging ξ_τ^G over all the niches: $\xi^L = \frac{1}{P_G}\sum_{\tau=1}^{P_G}\xi_\tau^G$.

AUTHOR CONTRIBUTIONS

SH, SS-M, AF, and MA: conceived the experiments, analyzed the data, and wrote and revised the manuscript. SH and MA: designed the experiments. SH: designed the software used in the analysis.

FUNDING

MA thanks the Marcos Moshinsky Foundation for the 2014 Fellowship. SH thanks CONACyT for a Ph.D. scholarship. This work was partly supported by PAPIT-UNAM grant IN226917.

ACKNOWLEDGMENTS

We thank Philippe Cluzel, Osbaldo Resendis and Pablo Vinuesa for useful comments and discussions.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphys.2018.01836/full#supplementary-material

REFERENCES

Albert, R., and Othmer, H. G. (2003). The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in drosophila melanogaster. *J. Theor. Biol.* 223, 1–18. doi: 10.1016/S0022-5193(03)00035-3

Aldana, M. (2003). Boolean dynamics of networks with scale-free topology. *Physica D* 185, 45–66. doi: 10.1016/S0167-2789(03)00174-X

Aldana, M., Balleza, E., Kauffman, S., and Resendiz, O. (2007). Robustness and evolvability in genetic regulatory networks. *J. Theor. Biol.* 245, 433–448. doi: 10.1016/j.jtbi.2006.10.027

Andrew, W., Kohl, K. D., Brucker, R. M., Edward, J., Opstal, V., Phylosymbiosis, S. R. B., et al. (2016). Phylosymbiosis : relationships and functional effects of microbial communities across host evolutionary history. *PLoS Biol.* 14:e2000225. doi: 10.1371/journal.pbio.2000225

Balleza, E., Alvarez-Buylla, E. R., Chaos, A., Kauffman, S., Shmulevich, I., and Aldana, M. (2008). Critical dynamics in genetic regulatory networks: examples from four kingdoms. *PLoS ONE* 3:e2456. doi: 10.1371/journal.pone.0002456

Bates, J. M., Mittge, E., Kuhlman, J., Baden, K. N., Cheesman, S. E., and Guillemin, K. (2006). Distinct signals from the microbiota promote different aspects of zebrafish gut differentiation. *Dev. Biol.* 297, 374–386. doi: 10.1016/j.ydbio.2006.05.006

Blaser, M. J. (2017). The theory of disappearing microbiota and the epidemics of chronic diseases. *Nat. Rev. Immunol.* 17, 461–463. doi: 10.1038/nri.2017.77

Blaser, M. J., and Falkow, S. (2009). What are the consequences of the disappearing human microbiota? *Nat. Rev. Microbiol.* 7, 887–894. doi: 10.1038/nrmicro2245

Bordenstein, S. R., and Theis, K. R. (2015). Host biology in light of the microbiome: ten principles of holobionts and hologenomes. *PLoS Biol.* 13:e1002226. doi: 10.1371/journal.pbio.1002226

Brucker, R. M., and Bordenstein, S. R. (2013). The hologenomic basis of speciation: gut bacteria cause hybrid lethality in the genus nasonia. *Science* 466, 667–669. doi: 10.1126/science.1240659

Camp, J. G., Frank, C. L., Lickwar, C. R., Guturu, H., Rube, T., Wenger, A. M., et al. (2014). Microbiota modulate transcription in the intestinal epithelium without remodeling the accessible chromatin landscape. *Genome Res.* 24, 1504–1516. doi: 10.1101/gr.165845.113

Cho, I., and Blaser, M. J. (2012). The human microbiome: at the interface of health and disease. *Nat. Rev. Genet.* 13, 260–270. doi: 10.1038/nrg3182

Consortium, H. M. P. (2012). Structure, function and diversity of the healthy human microbiome. *Nature* 486, 207–214. doi: 10.1038/nature11234

Contreras, A. V., Cocom-Chan, B., Hernandez-Montes, G., Portillo-Bobadilla, T., and Resendis-Antonio, O. (2016). Host-microbiome interaction and cancer: potential application in precision medicine. *Front. Physiol.* 7:606. doi: 10.3389/fphys.2016.00606

Coyte, K. Z., Schluter, J., and Foster, K. R. (2015). The ecology of the microbiome: networks, competition, and stability. *Science* 350, 663–666. doi: 10.1126/science.aad2602

Daniels, B. C., Kim, H., Moore, D., Zhou, S., Smith, H. B., Karas, B., et al. (2018). Criticality distinguishes the ensemble of biological regulatory networks. *Phys. Rev. Lett.* 121, 138102. doi: 10.1103/PhysRevLett.121.138102

Davidich, M. I., and Bornholdt, S. (2008). Boolean network model predicts cell cycle sequence of fission yeast. *PLoS ONE* 3:e1672. doi: 10.1371/journal.pone.0001672

Davidson, E. H. (2010). Emerging properties of animal gene regulatory networks. *Nature* 468, 911. doi: 10.1038/nature09645

Davidson, E. H., and Levine, M. S. (2008). Properties of developmental gene regulatory networks. *Proc. Natl. Acad. Sci. U.S.A.* 105, 20063–20066. doi: 10.1073/pnas.0806007105

Deines, P., Lachnit, T., and Bosch, T. C. (2017). Competing forces maintain the hydra metaorganism. *Immunol. Rev.* 279, 123–136. doi: 10.1111/imr.12564

Derrida, B., and Pomeau, Y. (1986). Random networks of automata: a simple annealed approximation. *Europhys. Lett.* 1, 45–49. doi: 10.1209/0295-5075/1/2/001

Donaldson, G. P., Lee, S. M., and Mazmanian, S. K. (2015). Gut biogeography of the bacterial microbiota. *Nat. Rev. Microbiol.* 14, 20–32. doi: 10.1038/nrmicro3552

Doolittle, W. F., and Booth, A. (2017). It's the song, not the singer: an exploration of holobiosis and evolutionary theory. *Biol. Philos.* 32, 5–24. doi: 10.1007/s10539-016-9542-2

Doolittle, W. F., and Inkpen, S. A. (2018). Processes and patterns of interaction as units of selection: an introduction to ITSNTS thinking. *Proc. Natl. Acad. Sci. U.S.A.* 115, 201722232. doi: 10.1073/pnas.1722232115

Douglas, A. E., and Werren, J. H. (2016). Holes in the hologenome: why host-microbe symbioses are not holobionts. *MBio* 7, e02099–15. doi: 10.1128/mBio.02099-15

Ereshefsky, M., and Pedroso, M. (2015). Rethinking evolutionary individuality. *Proc. Natl. Acad. Sci. U.S.A.* 112, 10126–10132. doi: 10.1073/pnas.1421377112

Espinosa-Soto, C., Padilla-Longoria, P., and Alvarez-Buylla, E. R. (2004). A gene regulatory network model for cell-fate determination during arabidopsis thaliana flower development that is robust and recovers experimental gene expression profiles. *Plant Cell* 16, 2923–2939. doi: 10.1105/tpc.104.021725

Farrell, J. J., Zhang, L., Zhou, H., Chia, D., Elashoff, D., Akin, D., et al. (2011). Variations of oral microbiota are associated with pancreatic diseases including pancreatic cancer. *Gut* 61, 582–588. doi: 10.1136/gutjnl-2011-300784

Fernández, L., Langa, S., Martín, V., Maldonado, A., Jiménez, E., Martín, R., et al. (2013). The human milk microbiota: origin and potential roles in health and disease. *Pharmacol. Res.* 69, 1–10. doi: 10.1016/j.phrs.2012.09.001

Fitzpatrick, B. M. (2014). Symbiote transmission and maintenance of extra-genomic associations. *Front. Microbiol.* 5, 1–15. doi: 10.3389/fmicb.2014.00046

Foster, K. R., and Bell, T. (2012). Competition, not cooperation, dominates interactions among culturable microbial species. *Curr. Biol.* 22, 1845–1850. doi: 10.1016/j.cub.2012.08.005

Francescone, R., Hou, V., and Grivennikov, S. I. (2014). Microbiome, inflammation and cancer. *Cancer J.* 20, 181. doi: 10.1097/PPO.0000000000000048

Gilbert, S. F., Sapp, J., and Tauber, A. I. (2012). A symbiotic view of life: we have never been individuals. *Q. Rev. Biol.* 87, 325–341. doi: 10.1086/668166

Gonzalez, A., Stombaugh, J., Lozupone, C., Turnbaugh, P. J., Gordon, J. I., and Knight, R. (2011). The mind-body-microbial continuum. *Dialogues Clin. Neurosci.* 13, 55–62.

Gordon, J., Knowlton, N., Relman, D. A., Rohwer, F., and Youle, M. (2013). Superorganisms and holobionts. *Microbe* 8, 152–153. doi: 10.1128/microbe.8.152.1

Grice, E. A., and Segre, J. A. (2011). The skin microbiome. *Nat. Rev. Microbiol.* 9, 244–253. doi: 10.1038/nrmicro2537

Guerrero, R., Margulis, L., and Berlanga, M. (2013). Symbiogenesis: the holobiont as a unit of evolution. *Int. Microbiol.* 16, 133–143. doi: 10.2436/20.1501.01.188

Guo, L., Zhao, G., Xu, J. R., Kistler, H. C., Gao, L., and Ma, L.-J. (2016). Compartmentalized gene regulatory network of the pathogenic fungus fusarium graminearum. *New Phytol.* 211, 527–541. doi: 10.1111/nph.13912

Halfvarson, J., Brislawn, C. J., Lamendella, R., Vázquez-Baeza, Y., Walters, W. A., Bramer, L. M., et al. (2017). Dynamics of the human gut microbiome in inflammatory bowel disease. *Nat. Microbiol.* 2, 1–7. doi: 10.1038/nmicrobiol.2017.4

Hameya, F. K., Nestorowaa, S., Kinstona, S. J., Kenta, D. G., Wilsona, . N. K., and Göttgens, B. (2017). Reconstructing blood stem cell regulatory network models from single-cell molecular profiles. *Proc. Natl. Acad. Sci. U.S.A.* 114, 5822–5829. doi: 10.1073/pnas.1610609114

Hooper, L. V., Littman, D. R., and Macpherson, A. J. (2012). Interactions between the microbiota and the immune system. *Science* 336, 1268–1273. doi: 10.1126/science.1223490

Hooper, L. V., Wong, M. H., Thelin, A., Hansson, L., Falk, P. G., and Gordon, J. I. (2001). Molecular analysis of commensal host-microbial relationships in the intestine. *Science* 291, 881–884. doi: 10.1126/science.291.5505.881

Huang, S., Eichler, G., Bar-Yam, Y., and Ingber, D. E. (2005). Cell fates as high-dimensional attractor states of a complex gene regulatory network. *Phys. Rev. Lett.* 94, 128701. doi: 10.1103/PhysRevLett.94.128701

Kauffman, S. (1969a). Homeostasis and differentiation in random genetic control networks. *Nature* 224, 177–178. doi: 10.1038/224177a0

Kauffman, S. A. (1969b). Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.* 22, 437–467. doi: 10.1016/0022-5193(69)90015-0

Kinouchi, O., and Copelli, M. (2006). Optimal dynamical range of excitable networks at criticality. *Nat. Phys.* 2, 348–351. doi: 10.1038/nphys289

Krajmalnik-Brown, R., Ilhan, Z.-E., Kang, D.-W., and DiBaise, J. K. (2012). Effects of gut microbes on nutrient absorption and energy regulation. *Nutr. Clin. Pract.* 27, 201–214. doi: 10.1177/0884533611436116

Laland, K., Uller, T., Feldman, M., Sterelny, K., Müller, G. B., Moczek, A., et al. (2014). Does evolutionary theory need a rethink? *Nature* 514, 161–164. doi: 10.1038/514161a

Langton, C. G. (1990). Computation at the edge of chaos: phase transitions and emergent computation. *Physica D* 42, 12–37. doi: 10.1016/0167-2789(90)90064-V

Ley, R. E. (2010). Obesity and the human microbiome. *Curr. opin. Gastroenterol.* 26, 5–11. doi: 10.1097/MOG.0b013e328333d751

Ley, R. E., Hamady, M., Lozupone, C., Turnbaugh, P. J., Ramey, R. R., Bircher, J. S., et al. (2008). Evolution of mammals and Their Gut Microbes. *Science* 320, 1647–1651. doi: 10.1126/science.1155725

Ley, R. E., Peterson, D. A., and Gordon, J. I. (2006a). Ecological and evolutionary forces shaping microbial diversity in the human intestine. *Cell* 124, 837–848. doi: 10.1016/j.cell.2006.02.017

Ley, R. E., Turnbaugh, P. J., Klein, S., and Gordon, J. I. (2006b). Microbial ecology: human gut microbes associated with obesity. *Nature* 444, 1022. doi: 10.1038/4441022a

Li, F., Long, T., Lu, Y., Ouyang, Q., and Tang, C. (2004). The yeast cell-cycle network is robustly designed. *Proc. Natl. Acad. Sci. U.S.A.* 101, 4781–4786. doi: 10.1073/pnas.0305937101

Lloyd-Price, J., Abu-Ali, G., and Huttenhower, C. (2016). The healthy human microbiome. *Genome Med.* 8, 1–11. doi: 10.1186/s13073-016-0307-y

Lynch, M. (2010). Evolution of the mutation rate. *Trends Genet.* 26, 345–352. doi: 10.1016/j.tig.2010.05.003

Lynch, M., Ackerman, M. S., Gout, J. F., Long, H., Sung, W., Thomas, W. K., et al. (2016). Genetic drift, selection and the evolution of the mutation rate. *Nat. Rev. Genet.* 17, 704–714. doi: 10.1038/nrg.2016.104

Ma, S., Snyder, M., and Dinesh-Kumar, S. P. (2017). Discovery of novel human gene regulatory modules from gene co-expression and promoter motif analysis. *Sci. Rep.* 7, 5557. doi: 10.1038/s41598-017-05705-2

Mai, V. (2004). Dietary modification of the intestinal microbiota. *Nutr. Rev.* 62, 235–242. doi: 10.1111/j.1753-4887.2004.tb00045.x

Manor, O., Levy, R., and Borenstein, E. (2014). Mapping the inner workings of the microbiome: genomic- and metagenomic-based study of metabolism and metabolic interactions in the human microbiome. *Cell Metab.* 20, 742–752. doi: 10.1016/j.cmet.2014.07.021

McCabe, L., Britton, R. A., and Parameswaran, N. (2015). Prebiotic and probiotic regulation of bone health: role of the intestine and its microbiome. *Curr. Osteoporos. Rep.* 13, 363–371. doi: 10.1007/s11914-015-0292-x

Moeller, A. H., Li, Y., Ngole, E. M., Ahuka-Mundeke, S., Lonsdorf, E. V., Pusey, A. E., et al. (2014). Rapid changes in the gut microbiome during human evolution. *Proc. Natl. Acad. Sci. U.S.A.* 111, 16431–16435. doi: 10.1073/pnas.1419136111

Moran, N. A., and Sloan, D. B. (2015). The hologenome concept: helpful or hollow? *PLoS Biol.* 13:e1002311. doi: 10.1371/journal.pbio.1002311

Morgan, X. C., Tickle, T. L., Sokol, H., Gevers, D., Devaney, K. L., Ward, D. V., et al. (2012). Dysfunction of the intestinal microbiome in inflammatory bowel disease and treatment. *Genome Biol.* 13, R79. doi: 10.1186/gb-2012-13-9-r79

Morris, J. J., Lenski, R. E., and Zinser, E. R. (2012). The black queen hypothesis: evolution of dependencies through adaptative gene loss. *Mbio* 3, e00036–12. doi: 10.1128/mBio.00036-12

Motta, S. S., Cluzel, P., and Aldana, M. (2015). Adaptive resistance in bacteria requires epigenetic inheritance, genetic noise, and cost of efflux pumps. *PLoS ONE* 10:e0118464. doi: 10.1371/journal.pone.0118464

Nicholson, J. K., Holmes, E., Kinross, J., Burcelin, R., Gibson, G., Jia, W., et al. (2012). Host-gut microbiota metabolic interactions. *Science* 336, 1262–1268. doi: 10.1126/science.1223813

Nykter, M., Price, N. D., Larjo, A., Aho, T., Kauffman, S. A., Yli-Harja, O., et al. (2008). Critical networks exhibit maximal information diversity in structure-dynamics relationships. *Phys. Rev. Lett.* 100, 058702. doi: 10.1103/PhysRevLett.100.058702

Oliveri, P., Tu, Q., and Davidson, E. H. (2008). Global regulatory logic for specification of an embryonic cell lineage. *Proc. Natl. Acad. Sci. U.S.A.* 105, 5955–5962. doi: 10.1073/pnas.0711220105

Pérez-Landero, S., Sandoval-Motta, S., Martínez-Anaya, C., Yang, R., Folch-Mallol, J. L., Martínez, L. M., et al. (2015). Complex regulation of hsf1-skn7 activities by the catalytic subunits of pka in saccharomyces cerevisiae: experimental and computational evidences. *BMC Syst. Biol.* 9:42. doi: 10.1186/s12918-015-0185-8

Queller, D. C., and Strassmann, J. E. (2016). Problems of multi-species organisms: endosymbionts to holobionts. *Biol. Philos.* 31, 855–873. doi: 10.1007/s10539-016-9547-x

Rawls, J. F., Samuel, B. S., and Gordon, J. I. (2004). Gnotobiotic zebrafish reveal evolutionarily conserved responses to the gut microbiota. *Proc. Natl. Acad. Sci. U.S.A.* 101, 4596–4601. doi: 10.1073/pnas.0400706101

Resendis-Antonio, O., Hernández, M., Mora, Y., and Encarnación, S. (2012). Functional modules, structural topology, and optimal activity in metabolic networks. *PLoS Comput. Biol.* 8:e1002720. doi: 10.1371/journal.pcbi.1002720

Rogers, G., Keating, D., Young, R., Wong, M., Licinio, J., and Wesselingh, S. (2016). From gut dysbiosis to altered brain function and mental illness: mechanisms and pathways. *Mole. Psychiatry* 21, 738. doi: 10.1038/mp.2016.50

Rosenberg, E., Sharon, G., Atad, I., and Zilber-Rosenberg, I. (2010). The evolution of animals and plants via symbiosis with microorganisms. *Environ. Microbiol. Rep.* 2, 500–506. doi: 10.1111/j.1758-2229.2010.00177.x

Rosenberg, E., and Zilber-Rosenberg, I. (2016). Microbes drive evolution of animals and plants: the hologenome concept. *mBio* 7, e01395–15. doi: 10.1128/mBio.01395-15

Roughgarden, J., Gilbert, S. F., Rosenberg, E., Zilber-Rosenberg, I., and Lloyd, E. A. (2018). Holobionts as units of selection and a model of their population dynamics and evolution. *Biol. Theory* 13, 44–65. doi: 10.1007/s13752-017-0287-1

Rupnik, M., Wilcox, M. H., and Gerding, D. N. (2009). Clostridium difficile infection: New developments in epidemiology and pathogenesis. *Nat. Rev. Microbiol.* 7, 526–536. doi: 10.1038/nrmicro2164

Sachs, J. L., and Hollowell, a. C. (2012). The origins of cooperative bacterial communities. *mBio* 3, e00099–12. doi: 10.1128/mBio.00099-12

Sachs, J. L., Skophammer, R. G., Bansal, N., and Stajich, J. E. (2014). Evolutionary origins and diversification of proteobacterial mutualists. *Proc. R. Soc. B Biol. Sci.* 281, 20132146. doi: 10.1098/rspb.2013.2146

Sagan, L. (1967). On the origin of mitosing cells. *J. Theor. Biol.* 14, 225IN1–274IN6.

Sandoval-Motta, S., Aldana, M., and Frank, A. (2017). Evolving ecosystems: Inheritance and selection in the light of the microbiome. *Arch. Med. Res.* 48, 780–789. doi: 10.1016/j.arcmed.2018.01.002

Sears, C. L., and Garrett, W. S. (2014). Microbes, microbiota, and colon cancer. *Cell Host Microbe* 15, 317–328. doi: 10.1016/j.chom.2014.02.007

Seekatz, A. M., and Young, V. B. (2014). Clostridium difficile and the microbiota. *J. Clin. Invest.* 124, 4182–4189. doi: 10.1172/JCI72336

Serra, R., Villani, M., Graudenzi, A., and Kauffman, S. (2007). Why a simple model of genetic regulatory networks describes the distribution of avalanches in gene expression data. *J. Theor. Biol.* 246, 449–460. doi: 10.1016/j.jtbi.2007.01.012

Sharpton, T. J. (2018). Role of the gut microbiome in vertebrate evolution. *mSystems* 3:e00174-17. doi: 10.1128/mSystems.00174-17

Shin, S. C., Kim, S.-H., You, H., Kim, B., Kim, A. C., Lee, K.-A., et al. (2011). Drosophila microbiome modulates host developmental and metabolic homeostasis via insulin signaling. *Science* 334, 670–674. doi: 10.1126/science.1212782

Shmulevich, I., Kauffman, S. A., and Aldana, M. (2005). Eukaryotic cells are dynamically ordered or critical but not chaotic. *Proc. Natl. Acad. Sci. U.S.A.* 102, 13439–13444. doi: 10.1073/pnas.0506771102

Spor, A., Koren, O., and Ley, R. (2011). Unravelling the effects of the environment and host genotype on the gut microbiome. *Nat. Rev. Microbiol.* 9, 279. doi: 10.1038/nrmicro2540

Stern, M. D. (1999). Emergence of homeostasis and "noise imprinting" in an evolution model. *Proc. Natl. Acad. Sci. U.S.A.* 96, 10746–10751.

Taxis, T. M., Wolff, S., Gregg, S. J., Minton, N. O., Zhang, C., Dai, J., et al. (2015). The players may change but the game remains: network analyses of ruminal microbiomes suggest taxonomic differences mask functional similarity. *Nucleic Acids Res.* 43, 9600–9612. doi: 10.1093/nar/gkv973

Theis, K. R., Dheilly, N. M., Klassen, J. L., Brucker, R. M., Baines, J. F., Bosch, T. C. G., et al. (2016). Getting the hologenome concept right: an eco-evolutionary framework for hosts and their microbiomes. *mSystems* 1, e00028–16. doi: 10.1128/mSystems.00028-16

Torres-Sosa, C., Huang, S., and Aldana, M. (2012). Criticality is an emergent property of genetic networks that exhibit evolvability. *PLoS Comput. Biol.* 8:e1002669. doi: 10.1371/journal.pcbi.1002669

Tropini, C., Earle, K. A., Huang, K. C., and Sonnenburg, J. L. (2017). The gut microbiome: connecting spatial organization to function. *Cell Host Microbe* 21, 433–442. doi: 10.1016/j.chom.2017.03.010

Turnbaugh, P. J., and Stintzi, A. (2011). Human health and disease in a microbial world. *Front. Microbiol.* 2:190. doi: 10.3389/fmicb.2011.00190

van Nood, E., Vrieze, A., Nieuwdorp, M., Fuentes, S., Zoetendal, E. G., de Vos, W. M., et al. (2013). Duodenal infusion of donor feces for recurrent clostridium difficile. *New Engl. J. Med.* 368, 407–415. doi: 10.1056/NEJMoa1205037

van Opstal, E. J., and Bordenstein, S. R. (2015). Rethinking heritability of the microbiome. *Science* 349, 1172–1173. doi: 10.1126/science.aab3958

Wagner, G. P. (2007). The developmental genetics of homology. *Nat. Rev. Genet.* 8, 473–479. doi: 10.1038/nrg2099

West, S. A., Fisher, R. M., Gardner, A., and Kiers, E. T. (2015). Major evolutionary transitions in individuality. *Proc. Natl. Acad. Sci. U.S.A.* 112, 10112–10119. doi: 10.1073/pnas.1421402112

Wilson, I. D., and Nicholson, J. K. (2017). Gut microbiome interactions with drug metabolism, efficacy, and toxicity. *Transl. Res.* 179, 204–222. doi: 10.1016/j.trsl.2016.08.002

Yang, J., Tan, Q., Fu, Q., Zhou, Y., Hu, Y., Tang, S., et al. (2017). Gastrointestinal microbiome and breast cancer: correlations, mechanisms and potential clinical implications. *Breast Cancer* 24, 220–228. doi: 10.1007/s12282-016-0734-z

Yatsunenko, T., Rey, F. E., Manary, M. J., Trehan, I., Dominguez-Bello, M. G., Contreras, M., et al. (2012). Human gut microbiome viewed across age and geography. *Nature* 486, 222. doi: 10.1038/nature11053

Zackular, J. P., Baxter, N. T., Iverson, K. D., Sadler, W. D., Petrosino, J. F., Chen, G. Y., et al. (2013). The gut microbiome modulates colon tumorigenesis. *MBio* 4, e00692–13. doi: 10.1128/mBio.00692-13

Zilber-Rosenberg, I., and Rosenberg, E. (2008). Role of microorganisms in the evolution of animals and plants: the hologenome theory of evolution. *FEMS Microbiol. Revi.* 32, 723–735. doi: 10.1111/j.1574-6976.2008.00123.x

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

ORIGINAL RESEARCH
published: 21 January 2019
doi: 10.3389/fphys.2018.01964

Robustness of Nutrient Signaling Is Maintained by Interconnectivity Between Signal Transduction Pathways

Niek Welkenhuysen[1,2]*, *Barbara Schnitzer*[1,2], *Linnea Österberg*[1,2,3] *and Marija Cvijovic*[1,2]*

[1] Department of Mathematical Sciences, University of Gothenburg, Gothenburg, Sweden, [2] Department of Mathematical Sciences, Chalmers University of Technology, Gothenburg, Sweden, [3] Department of Biology and Biological Engineering, Chalmers University of Technology, Gothenburg, Sweden

Systems biology approaches provide means to study the interplay between biological processes leading to the mechanistic understanding of the properties of complex biological systems. Here, we developed a vector format rule-based Boolean logic model of the yeast *S. cerevisiae* cAMP-PKA, Snf1, and the Snf3-Rgt2 pathway to better understand the role of crosstalk on network robustness and function. We identified that phosphatases are the common unknown components of the network and that crosstalk from the cAMP-PKA pathway to other pathways plays a critical role in nutrient sensing events. The model was simulated with known crosstalk combinations and subsequent analysis led to the identification of characteristics and impact of pathway interconnections. Our results revealed that the interconnections between the Snf1 and Snf3-Rgt2 pathway led to increased robustness in these signaling pathways. Overall, our approach contributes to the understanding of the function and importance of crosstalk in nutrient signaling.

Keywords: nutrient signaling, cAMP-PKA pathway, Snf1 pathway, Snf3/Rgt2 pathway, logic modeling, Boolean logic model, crosstalk

OPEN ACCESS

Edited by:
Matteo Barberis,
University of Surrey, United Kingdom

Reviewed by:
Marco Vanoni,
Universitá degli studi di Milano Bicocca, Italy
Tomáš Helikar,
University of Nebraska-Lincoln, United States
Reka Albert,
Pennsylvania State University, United States

***Correspondence:**
Niek Welkenhuysen
niek@chalmers.se
Marija Cvijovic
marija.cvijovic@chalmers.se

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: *30 May 2018*
Accepted: *31 December 2018*
Published: *21 January 2019*

Citation:
Welkenhuysen N, Schnitzer B, Österberg L and Cvijovic M (2019) Robustness of Nutrient Signaling Is Maintained by Interconnectivity Between Signal Transduction Pathways. Front. Physiol. 9:1964. doi: 10.3389/fphys.2018.01964

1. INTRODUCTION

A biological system can be described as a set of components that interact in such a way that they form a functional unit (Alberghina and Westerhoff, 2005). Systems biology aims to understand the function of the components and how they interact at a systems level. This knowledge about the components provides predictability in the outcome of the system. However, the complexity of many biological processes obstructs the prediction of system outcomes. Mathematical modeling helps to compute the outcome of more complex systems and to identify the properties that emerge from the interaction between the components within the system. This can lead to an improved insight in the mechanistic properties of any biological system.

In signal transduction pathways components can undergo several different changes, such as phosphorylation on multiple sites that are further combined to achieve a subsequent reaction. These are very well-studied through both high-throughput and small scale studies making many components of signaling pathways known (Papin et al., 2005) and providing suitable data for utilizing systems biology approaches by developing a semi-quantitative logic (Boolean) models (Bornholdt, 2008; Wang et al., 2012).

To signal a broad spectrum of nutrients present in the cell environment the yeast *Saccharomyces cerevisiae* has an extensive nutrient sensing network in place. The function of this network is to initiate a comprehensive reprogramming of gene expression to be able to utilize specific nutrients. The yeast carbon and nitrogen sensing systems have been thoroughly studied and their key components have been identified (Gancedo, 2008; Broach, 2012; Conrad et al., 2014; Shashkova et al., 2015; Sanz et al., 2016). However, it is not sufficient just to know the components of a biological system. In order to gain a complete insight into the nutrient sensing system it is necessary to understand the functions of the components and how they interact with each other. In yeast, the carbon source sensing is mainly done by the cAMP-PKA pathway, Snf1 pathway, and the Snf3/Rgt2 pathway. Nitrogen source sensing is performed by the TOR pathway. The knowledge on the functioning of the components and the linearity of these pathways is ambiguous. The ambiguity is due to the substantial amount of crosstalk that has been identified between the components of the different pathways (Broach, 2012; Shashkova et al., 2015; Sanz et al., 2016).

Crosstalk, in biology, is a phenomenon by which an integrated intracellular signal from multiple inputs produces an output that is different from the response triggered by the individual pathways (Vert and Chory, 2011). Two pathways can be interconnected directly by shared component(s), or indirectly when one pathway affects another signaling pathway (Vert and Chory, 2011). The effect of crosstalk on signaling and regulatory pathways has already been studied through mathematical modeling, focusing on the crosstalk from kinases and phosphatases (Rowland et al., 2012, 2015; Rowland and Deeds, 2014). However, the action of kinases and phosphatases embedded in a full network (Endres, 2012) has not been deciphered. In this work we study the direct and indirect crosstalk between nutrient signaling pathways cAMP-PKA, Snf1, and Snf3/Rgt2. Experimental perturbation of these pathways produces noise causing a major challenge in identifying interconnections and therefore theoretical approaches, such as Boolean modeling, are often applied.

Boolean modeling has already been used to reconstruct various signaling pathways (Schlatter et al., 2009; Singh et al., 2012; Anderson et al., 2016). For nutrient sensing pathways a large network reconstruction of the Snf1 pathway has been made based on an exhaustive and manually curated literature review (Lubitz et al., 2015). Further, a logic model describing crosstalk between the Snf1 and Rgt2/Snf3 pathway has been published (Christensen et al., 2009). These however put the emphasis on the technical aspect of modeling of signaling pathways rather than on the predictive possibilities of the Boolean Model.

In this work we aimed to better understand if crosstalk within the yeast nutrient signaling network contributes to the vitality of the nutrient sensing function when the system is perturbed. Specifically, we look at how crosstalk between the Snf1, cAMP-PKA, and Rgt2/Snf3 pathways contribute to the appropriate response to nutritional availability. The model was transformed into a vector format rule-based Boolean model. The created model was completed and validated by a gap filling process based on known input/output relations. We further validated the model by experimental study of protein localization and phosphorylation status. This showed that the model can be used as a tool to predict states of components within the model. Next we included literature curated crosstalk between these pathways. The influence of the crosstalk on the network was evaluated through network perturbation and subsequent analysis of the component states. We found that some crosstalk reactions were vital for the functioning of the network. It was suggested that even in the non-perturbed state they played an important role. Other crosstalk reactions did not have any significant influence on the network output. We further show the modularity of our modeling approach by adding the nitrogen sensing TOR pathway to the model. Overall, we present a Boolean model of a large nutrient signaling network that allows to assess the influence of crosstalk on the network.

2. MATERIALS AND METHODS

2.1. Logic Model

The model of the nutrient sensing network was based on peer-published literature and each module in the code is denoted with the respective PubMedID of the article (Celenza and Carlson, 1989; Broach, 1991, 2012; Mitts et al., 1991; Kuroda et al., 1993; Haney and Broach, 1994; Hu et al., 1995; Ozcan and Johnston, 1995; Treitel and Carlson, 1995; Martinez-Pastor et al., 1996; Ozcan et al., 1996; Schmitt and McEntee, 1996; Colombo et al., 1998, 2004; Gorner et al., 1998; Lutfiyya et al., 1998; Frolova et al., 1999; Pedruzzi et al., 2000; Schmidt and McCartney, 2000; Jacinto et al., 2001; Düvel et al., 2003; Flick et al., 2003; Kim et al., 2003; Mosley et al., 2003; Cameroni et al., 2004; Moriya and Johnston, 2004; De Wever et al., 2005; Hong et al., 2005; Palomino et al., 2005; Roosen et al., 2005; Swinnen et al., 2006; Peeters et al., 2007; Lee et al., 2008, 2011, 2013; Rubenstein et al., 2008; Georis et al., 2009; Tate et al., 2010; Loewith and Hall, 2011; Orzechowski Westholm et al., 2012; Bontron et al., 2013; Hughes Hallett et al., 2014; Ma et al., 2014; Kayikci and Nielsen, 2015; Shashkova et al., 2017). The model (**Figure 1B**) was translated to a Boolean logic model and implemented in MATLAB© (The MathWorks, Inc.). In our model there are three types of components: metabolites, proteins and complex components. Each protein is assigned a state vector with six entries defining its name, presence, localization, phosphorylation status, GDP/GTP exchange status, and DNA binding status. A component can: (A) be present or absent, (B) be localized to the membrane, the cytosol or the nucleus, (C) have phosphorylation or guanosine groups, and (D) be bound to DNA. The second type of component, metabolites, are treated in the same manner, however, they only need three properties and therefore their state vector has only length three. Here, phosphorylation, GDP/GTP exchange, and DNA binding are redundant. In some reactions protein complexes are formed. Those are denoted by complex formation components with vector length one and indicate if the complex is active or not.

In the implementation all parameters in the state vector are translated to a bound set of integer values (**Tables S3**, **S4**), which are not necessarily purely Boolean but can include more possible outcomes. Each vector uniquely represents one state in the set

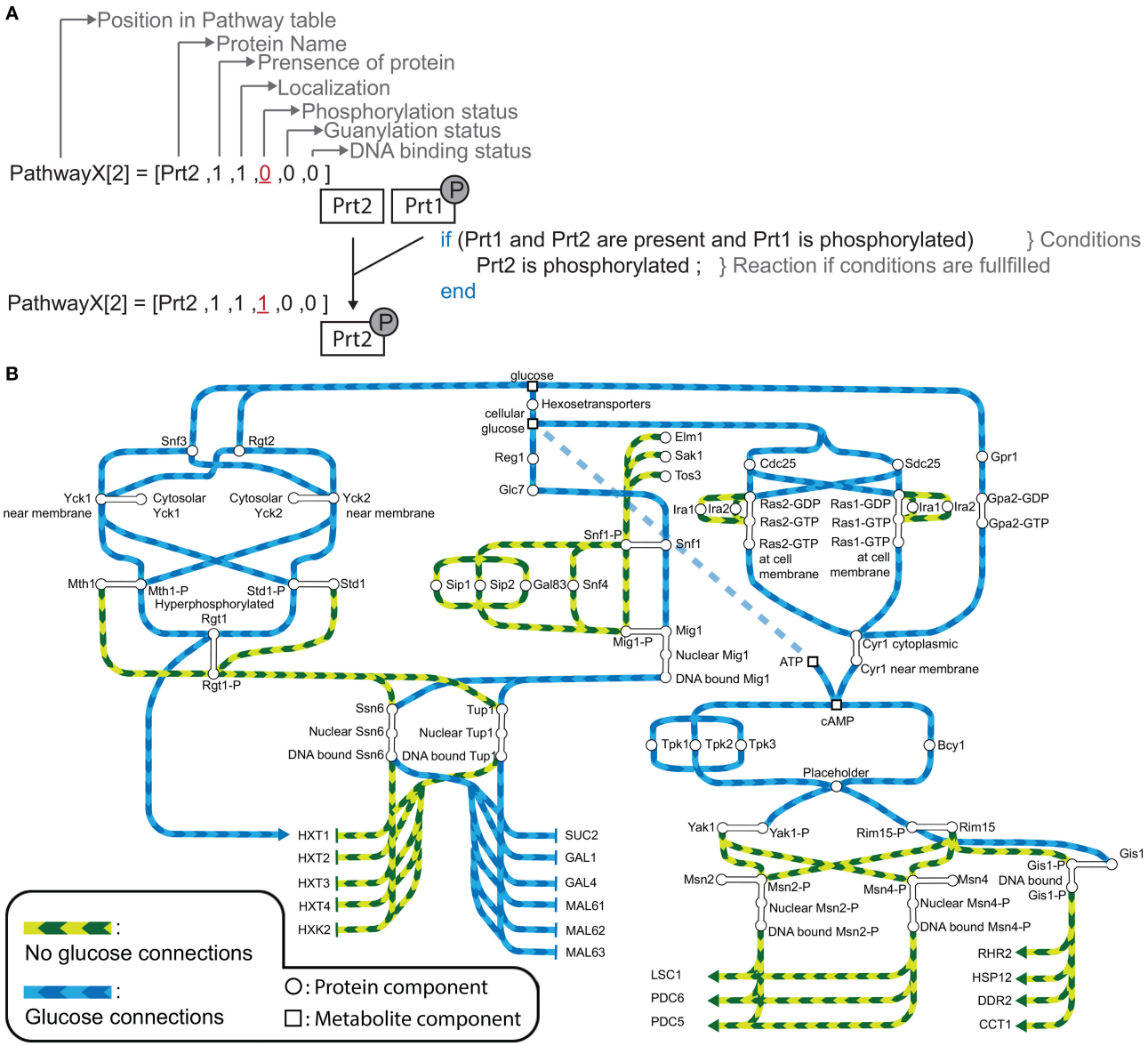


FIGURE 1 | (A) An example of a reaction in our model for hypothetical PathwayX, which contains hypothetical components Prt1 and Prt2. Each component is designated a vector, which is a collection of the name and the different states a component can assume. Components Prt1 and Prt2 belong to the table for Pathway X. A reaction in the model only occurs when the conditions described in the if-statement are fulfilled. **(B)** Signal transmission route of "glucose" and "no glucose" conditions through the Rgt2/Snf3, Snf1, and cAMP-PKA pathway and its components. The graph displays the possible states of each component for the system without perturbations (WT-system). Blue lines display the connections between the components when glucose is available. Green lines display the connections between the components when glucose is not available. Round nodes are proteins, square nodes display metabolites, upper-case letters are promoters of genes (e.g., SUC2). An arrow at the end of the lines displays activation of a gene and a vertical stripe gene repression. The connected nodes are the possible states one component can assume.

of all possible states. The components are ordered according to the pathway they belong to (**Tables S2**, **S6**). In total, the model comprises 4 metabolites, 63 proteins (including 6 unknown) in 4 pathways, and 19 target genes.

The initial model inputs are the metabolites glucose and nitrogen that can be set to present (1) or absent (0). Starting from that assumption, the information propagates through the pathways by numerous logical operations constructed based on the literature review. Biologically, most modifications are equivalent to activation or inhibition through phosphorylation/dephosphorylation or GDP/GTP exchange. **Figure 1A** shows an example for an operation involving two proteins in an arbitrary pathway XXXpw: if protein 2 is present (XXXpw{2,2} == 1) AND protein 1 is present (XXXpw{1,2} == 1) AND phosphorylated

(XXXpw{1,4} == 1), protein 2 gets phosphorylated (XXXpw{2,4} = 1). The phosphorylation status of protein 2 therefore increases from 0 to 1.

In the model a typical operation is therefore a change in the state vector of a component that only happens under certain conditions (rules for an reaction to happen). Conditions are usually composed of one or more state requirements that are connected with logical operators AND or OR. States can only alter within the defined state space presented in **Tables S3**, **S4**. All reactions in the pathways that were implemented are executed asynchronously. Therefore, an induced state change has immediate effects on the next steps in the model. The algorithm stops if no operation causes a state change in any component anymore, thus the logical steady state is reached. From this information it can be concluded which genes are active or not. In summary, the presence or absence of nutrients leads to a cascade of events and finally expression or repression of target genes.

The model can optionally simulate knockouts or deletions of components. It is equivalent to setting the component's "presence" state in the model to 0. Consequently, such a perturbed component cannot participate in any operation in the model. The eliminated components are listed by their names and given as input to the model. All pathways are connected by crosstalk that can be manipulated in the model. The crosstalk reactions, listed in **Table 2** and **Table S8**, can be switched on (1) and off (0) as a complementary input. By activating crosstalk, additional operations between proteins belonging to different pathways are appended.

The output is organized in tables sorted by pathways. In addition, separate tables are generated for the metabolites and for miscellaneous proteins that are shared over multiple pathways. Each component is part of exactly one table in which its steady state vector is given. Besides ordinary text files, a schematic picture of the cell for each pathway is created (**Figures 3A–D**, **4A**). Moreover, an extra file with all involved genes and their final status as the output of the model is saved.

Furthermore, the model is designed in such a way that it can sequentially switch between input metabolites, i.e., from no glucose to glucose or vice versa. Under each nutrient condition the steady state is found and used as an initial condition for the next iteration. Outputs are generated after each step. The MATLAB code of the model and the simulations is provided at https://github.com/cvijoviclab/LogicModel.

2.2. Yeast Strains and Culture

The *S. cerevisiae* yeast strains were grown overnight to mid-log phase at 30°C in Yeast Nitrogen Base (YNB) synthetic complete medium containing 1.7 g/l yeast nitrogen base, 5 g/l ammonium sulfate, 670 mg/l complete supplement mix supplied with the appropriate amount of carbon source. All used strains in this work are summarized in **Table S1**.

2.3. Fluorescence Microscopy

The overnight culture grown on YNB supplemented with 4% glucose was diluted to an OD of 0.5 in either YNB media supplemented with 4% glucose or 3% ethanol depending on which environmental conditions was imaged. Fluorescent images were obtained by capturing 5 μl media between a microscopic slide and a cover glass. This was inserted in an inverted Leica DMI4000 microscope with a Leica CTR 4000 fluorescent light source and Leica DMI4000 Bright field light source operating on the LAS AF operating system (AF6000 E). Images were acquired using a HCX PL APO CS 100.0X1.40 oil objective with the LECA DFC360 FX camera. Exposure times used were 20 ms for the bright field state, 320 ms for the red fluorescent (mCherry) state, and 350 ms for the green fluorescent state (GFP).

2.4. Western Blot

The *S. cerevisiae* yeast strain was grown overnight to mid-log phase at 30°C in YNB supplemented with 6% glucose. The cultures were diluted 1:2 with fresh YNB media supplemented by either 4% glucose or 0.05% and incubated for 2 h at 30°C. Five milliliters was used for sampling. NaOH was added to a final concentration of 0.1 M and left for incubation at room temperature for 5 min. The samples were spun down and the pellet resuspended in 400 μl of 2M NaOH with 7% beta mercaptoethanol and incubated for 2 min. at room temperature. Four hundred microliters of 50% TCA buffer was added and the samples were spun down. The pellet was washed with 500μl Tris-HCl, resuspended in 50 μl sample buffer [62.5 mM Tris-HCL(pH = 6.8), 3% SDS, 10% glycerol, 5% beta mercaptoethanol] and boiled for 5 min at 100°C. Protein concentration was determined using DC™ Protein Assay, BioRad. Thirty microliters of 6 mg/ml protein was loaded on a 4–20% Mini-PROTEAN® TGX Stain-Free™ Protein Gel, BioRad. The gel was imaged for full protein using Gel Doc EZ System, BioRad, and blotting was done using the Trans-Blot© Turbo™ Transfer System, BioRad. The membrane was washed 3 x 5 min with 20 ml TBS buffer before blocking and after incubation with the antibodies. Blocking was done for 1 h using Western Blocker™ Solution for HRP detection systems, Sigma-Aldrich. The membrane was incubated for 1 h 15 min with Phospho-AMPKa (Thr172) (40H9) Rabbit mAb, Cell Signaling, diluted 1:1,000 and 1 h with TidyBlot, BioRad diluted 1:500. The membrane was imaged using ChemiDoc™ Imaging Systems, BioRad and SuperSignal™ West Pico PLUS Chemiluminescent Substrate, Thermo Scientific™.

3. RESULTS

3.1. Vector Based Logic Modeling Allows for Modeling Protein States

Constructing the topologies of signaling networks is a challenging task, mainly because one protein can be in many different states, for example phosphorylation status and localization (Rother et al., 2013). In typical Boolean networks, nodes can only take the discrete values "0" and "1", meaning a node is either inactive or active, and if active the signal is passed on to the next node. This approach does not allow for discrimination between multiple states of a node without introducing new nodes that would represent each single state. The complexity of the system would in this way be vastly increased. Therefore, an approach is required that allows the nodes of the model to be in several states. A multi-valued logical model is able to take into account several states (Abou-Jaoudé

et al., 2016). However, this approach become impractical when there is a large amount of multiple states in which several states results in the same outcome. To overcome this obstacle we apply a vector format to a rules based model (Hlavacek et al., 2006; Boutillier et al., 2018). In a rule based model a reaction, defined as a state change of a node, only occurs given that certain exceeding rules or conditions are fulfilled. These conditions are defined as the required states of nodes for a reaction to be generated. Granted that no other reaction in the system will change this state, the node is in the logical steady state (LSS). We further assign every node, from here referred to as component, a component specific vector. In our modeling approach we distinguish between three different components: a protein component, metabolite component, and a complex formation component. The last component type is used for complex formation, and can only be "1" (active) or "0" (inactive).

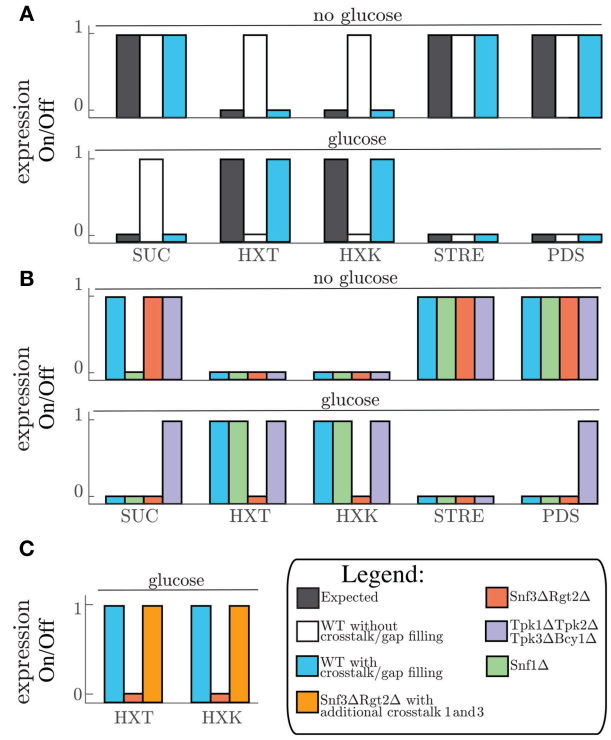


FIGURE 2 | (A) Expected gene expression pattern (black, left) compared to the predicted gene expression state from the model without (white, middle) and with addition of crosstalk reactions 7 and 9 (**Table 2**), and after the gap filling process (Blue, right) for "no glucose" conditions (upper part) and "glucose" conditions (lower part) given for all the grouped genes. **(B)** Predicted gene expression state for WT-model (crosstalk reactions 7 and 9, and gap filling process) (blue,left) and the perturbations *snf1*Δ (green, middle left−side), *snf3*Δ*rgt2*Δ (red, middle−right side), and *tpk1*Δ*tpk2*Δ*tpk3*Δ*bcy1*Δ (purple, right) given for all the grouped genes. **(C)** Predicted gene expression state for WT-model (blue, left) compared to predicted gene expression states of *snf3*Δ*rgt2*Δ without (red, middle) and with crosstalk reaction 1 and 3 (orange, right) for the gene group HXT and HXK. SUC is the name for the gene SUC2. HXT is the group name for genes HXT1, HXT2, HXT3, and HXT4. HXK is the name for the gene HXK2. STRE is the group name for LSC1, PDC6, and PDC5. PDS is the group name for RHR2, HSP12, DDR2, and CCT1.

For metabolite and protein components a different vector format is used (**Tables S2**, **S3**). The vector for a protein component has 6 positions which describe the name, presence, localization, phosphorylation status, GDP/GTP exchange status, and DNA binding status of the protein. In the metabolite vector there are 3 positions which describe name, presence, and localization of the metabolite. For example, hypothetical signaling pathway X consisting of protein components Prt1 and Prt2 with the system only having one condition (**Figure 1A**). When simulating the system component Prt2 is initially not phosphorylated, therefore, position four in the component vector is "0." When the conditions are fulfilled, namely both Prt1 and Prt2 are present in the system and Prt1 is phosphorylated, only then does position four in the vector for Prt2 change to "1", meaning that the protein becomes phosphorylated. We used this framework to reconstruct a model describing glucose signaling networks derived from literature. The reconstruction included the Snf3/Rgt2, the Snf1 pathway and the cAMP-PKA pathway (**Figure 1B**). We manually mined the literature to find the components needed to connect the input conditions ("glucose" or "no glucose") to the output gene expression. For yeast, glucose is a preferred carbon source since it can enter directly into the glycolysis after import into the cell. Therefore, yeast will prefer to metabolize glucose over other carbon sources. This model encompasses 48 components of which 45 are protein components and 3 are metabolite components (**Table S2**). All of these are unique proteins and metabolites except for the hexose transporters. Transporters Hxt1 to Hxt17 are a group of hexose transporters of which each has different glucose uptake characteristics (Kruckeberg, 1996; Horak, 2013). To reduce the complexity, we have grouped them together as one protein component named HXTs. The Rgt1 transcription factor becomes hyper-phosphorylated when the cell is exposed to glucose and is phosphorylated in a minor extent when glucose is not available (Flick et al., 2003). Therefore, we have chosen to assign the status of hyper-phosphorylated Rgt1 as "1" and "0" for the minor phosphorylated status in the component vector on the position for phosphorylation status. All the components in the model are divided into five different tables: metabolites, Snf1pw, R2S3Pathway, PKApw, and Miscl. The last table, Miscl, is for the metabolites and components of the Snf1 pathway, Rgt2/Snf3 pathway, cAMP-PKA pathway, and protein components belonging to neither or being shared over more than one of the previously named pathways. These tables are comprised of the component vectors. Further the model includes one complex component to signal the formation of an active PKA complex. Overall, the components take part in 61 rules or conditions (**Table S5**). This model reconstruction gives an overview of the connections between the involved components in glucose signaling reactions.

3.2. Gap Filling Processes Reveal a Lack of Protein Phosphatase Components and the Importance of Crosstalk From PKA Pathway

From this model we set out to make a system that can switch between "glucose" and "no glucose" as input conditions and

make it reproduce the correct RNA expression profile as an output. To validate this we let the model reach the LSS for a certain condition after initialization and thereafter switch to the other condition. The original model from the literature reconstruction (**Figure 1B**) was able to correctly simulate the LSS for the first input conditions but unable to switch to the second expected LSS (**Figure 2A**). We therefore used the simulation to analyze what steps in the network are missing to successfully simulate the expected outcome. Additional unknown components needed to be added in the model to compensate for missing reactions that eventually lead to correct RNA expression profiles in all cases. This gap filling process was performed in an iterative model extension process suggested in an earlier study on carbon signaling pathways (Lubitz et al., 2015). To successfully reproduce the input/output of the network we added six additional conditions (**Table 1**). This resulted in the addition of four unknown protein components to the model. These unknown components were added to the table of miscellaneous protein components (Miscl). Interestingly, the first four gaps required the addition of a protein phosphatase component. The finding that four out of six unknown parts that needed to be added to the model contained protein phosphatases is intriguing. This suggests a general lack of knowledge about dephosphorylation processes of proteins in the glucose signaling network. The other two parts required the addition of a known crosstalk reaction from the PKA pathway to the Rgt2/Snf3 and Snf1 pathways (**Table 2**).

TABLE 1 | Gap filling: Added parts after gap filling procedure in order to make the model switch between LSS for "glucose" and "no glucose" conditions.

#	Involved components	Gap description	Added component
1	Std1, Rgt1	Dephosphorylation of Std1 and Rgt1	Xxx1
2	Yak1, Rim15	Dephosphorylation of Yak1 and Rim5	Xxx2
3	Reg1, Glc7	Dephosphorylation of PP1 complex Reg1-Glc7	Xxx3
4	Msn2, Msn4	Dephosphorylation of Msn2 and Msn4	Xxx4
5	Glc7, Reg1	Phosphorylation of Glc7-Reg1	Crosstalk 7 (**Table 2**)
6	Rgt1	Phosphorylation of Rgt1	Crosstalk 9 (**Table 2**)

3.3. Vector Format Boolean Network Simulation Can Predict and Visualize the State of Network Components

After the gap filling process the model could simulate the switching between input conditions and predict the matching output status (**Figure 2A**). When predicting the outcome for one condition we initialize the model to the opposite condition first, since signaling networks are in place to sense changing conditions. Through model simulations we can test the effect of "glucose" and "no glucose" conditions on the model components. By plotting the component tables of these simulations in a graphical overview we create a coherent and legible way to view the pathway components and their different states (**Figure 3**). This neat overview simplifies comparison of the simulated LSS for the components with physical experiments. To show this feature we selected the transcription factors Msn2, Rgt1, and Mig1 to represent each pathway involved in glucose signaling and the general transcriptional repressors Tup1 and Ssn6. A version of these proteins, tagged with a fluorescent protein, was observed under the microscope in 4% glucose and in 3% ethanol as carbon source, representing "glucose" condition and "no glucose" condition respectively. Msn2, a transcription factor targeted by the PKA complex, localized to the nucleus with ethanol as carbon source and remained in the cytosol when exposed to glucose according to the model predictions (**Figure 3A** and **Figure S2**). When observing Msn2 labeled with a fluorescent green protein (GFP) in "glucose" conditions we detect a uniform distribution throughout the cell of the fluorescent signal from the GFP molecule. When the cells are grown in "no glucose" conditions the signal from the GFP molecule is no longer evenly distributed

TABLE 2 | Crosstalk: different types of crosstalk added to the model.

#	Involved components	Description	Source
1	Snf1, Mth1, Std1	Active Snf1 prevents inactivation of Mth1 and Std1	Gadura et al., 2006; Pasula et al., 2007
2	Snf1, Std1	Std1 stimulates the Snf1 kinase activity	Hubbard et al., 1994; Tomás-Cobos and Sanz, 2002; Kuchin et al., 2003
3	Reg1, Glc7, Yck1, Yck2	Reg1-Glc7 acts as an upstream activator of Yck1 and Yck2	Gadura et al., 2006
4	PKA complex, Sak1	PKA complex phosphorylates Sak1	Barrett et al., 2012
5	Snf1, PKA complex	PKA complex negatively regulates the Snf1 pathway (Sak1 independent)	Barrett et al., 2012
6	Snf1, Msn2	Snf1 can phosphorylate Msn2	De Wever et al., 2005
7	PKA complex, Glc7, Reg1	glucose activation of the PKA complex pathway is required for activation of PP1 (Glc7-Reg1)	Castermans et al., 2012
8	Snf1, Cyr1	Snf1 deactivates Cyr1 by phosphorylation	Nicastro et al., 2015
9	PKA complex, Rgt1	Bcy1 phosphorylates Rgt1 under high "glucose" conditions	Kim et al., 2006; Jouandot et al., 2011; Roy et al., 2013

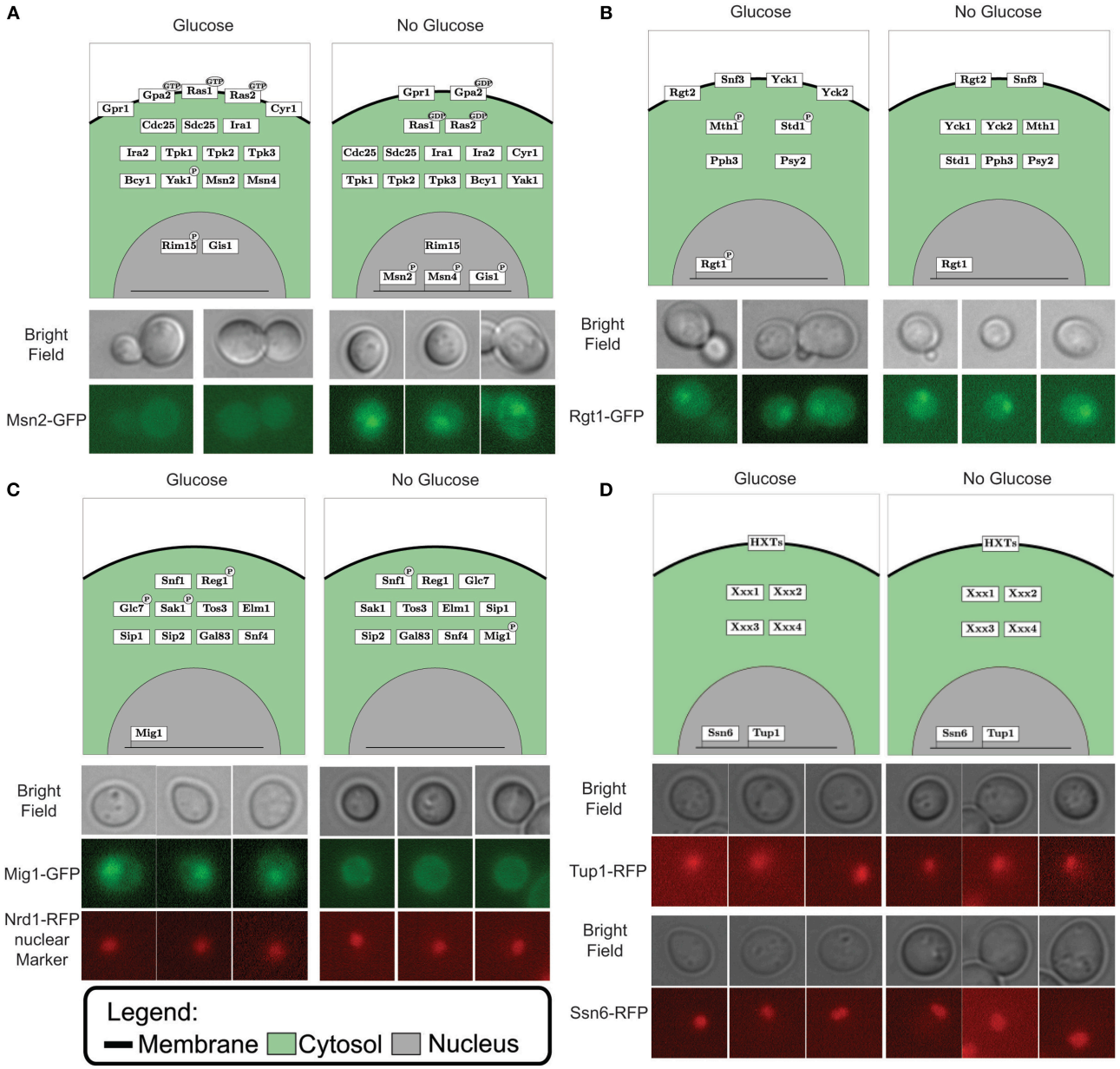


FIGURE 3 | Simulated localization vs. microscopy data for Msn2 in the PKA pathway **(A)**, Rgt1 in the Rgt2/Snf3 pathway **(B)**, Mig1 in the Snf1 pathway **(C)** and both Ssn6 and Tup1 as general transcription repressors **(D)**. The upper part of the each panel displays the graphical representation of the simulated vector tables for each pathway. Protein location is depicting by the box in either the membrane (black line), cytosol (green area) or nucleus (gray area). Post-translational modification such as GDP/GTP binding and phosphorylation are displayed by a black ellipse or circle on the protein box. The DNA bound status is give by the protein box being connected to the line (which displays DNA). The lower part of each panel displays the microscopy data. The images above display the bright field, the lower images displays the fluorescent marked transcription factor for **(A,B)**. The middle image display the fluorescent marked Mig1 transcription factor and the lower image display Nrd1 bound with a red fluorescent protein used a marker for the nucleus **(C)**. Panel **(D)** displays the bright field images in the upper panel and the fluorescently marked general transcription repressors Ssn6 and Tup1 in the lower part.

with the majority of signal focused in one part of the cell. This result indicates that Msn2 protein is localized in the nucleus. For Rgt1 the model prediction anticipates Rgt1 to be present in the nucleus for both "glucose" and "no glucose" conditions (**Figure 3B** and **Figure S2**). Because it either activates the HXT1 promoter in response to glucose availability (Mosley et al., 2003) or binds to the promoters of the hexose transporters to recruit transcription repressors when glucose is depleted (Kim et al., 2003; Broach, 2012). Observation of the yeast strain with GFP labeled Rgt1 showed that under both environmental conditions Rgt1 remained in the nucleus. As it has been shown in the literature and in our model predictions transcription factor Mig1 targeted by the Snf1 pathway. Mig1 is nuclear when the cell is exposed to glucose and remains in the cytosol when growing

on ethanol (De Vit et al., 1997) (**Figure 3C** and **Figure S2**). A yeast strain with both Mig1 tagged with GFP and Nrd1, a protein that always resides in the nucleus, bound to a red fluorescent protein (RFP) was used to determine the localization of Mig1. We observed that under "glucose" conditions Mig1 co-localizes to the Nrd1-RFP signal, but under "no glucose" conditions it remains uniformly distributed throughout the cell. The transcription repressor complex Ssn6-Tup1 is either recruited by Mig1 under "glucose" conditions or by Rgt1 when the cells are not exposed to glucose (Treitel and Carlson, 1995; Roy et al., 2013) (**Figure 3D** and **Figure S2**). Indeed, it was observed that under 4% glucose and 3% ethanol both Ssn6 and Tup1 are localized in the nucleus. In addition to component localization, the model also considers post-transcriptional modifications such as phosphorylation. The phosphorylation state can be used to validate the model. Dephosphorylation of the protein Snf1 occurs when the cells are exposed to glucose and Snf1 becomes phosphorylated when grown on ethanol as sole carbon source. Typically, phosphorylation status of proteins is measured by Western blot. When looking at the phosphorylation status of Snf1 via Western blot we observed that the Snf1 phosphorylation status from the model predictions and experimental results are similar (**Figure 3C** and **Figure S1**). In general, this shows that the model prediction can be validated not only with the RNA expression but also through observation of localization and post-transscriptional modification.

3.4. Crosstalk Reactions From cAMP-PKA to Rgt2/Snf3 can Restore Perturbed Network Signaling

The gap filling process showed that crosstalk reactions were required in order for the model to switch from one condition to another. We therefore collected known crosstalk reactions from the literature and selected 9 crosstalk reactions to test in our model (**Table 2**). Next, we looked for crosstalk combinations that contribute to the robustness of the yeast cell carbon source sensing system. The carbon source sensing system was perturbed for each pathway by removing (a) key protein component(s) from the model simulation (analogous to protein deletion). From her on the wild-type model with the gap filling parts (**Table 1**) and crosstalk reaction promoting PKA-dependent phosphorylation of Glc7 and Rgt1 will be referred as the wild-type (WT) model. We always included these additions in the WT simulations since they were crucial to have the correct expected gene expression profile as simulation outcome (**Figure 2A**). When referred to the WT model we mean the model in which no protein components are removed from the simulation. Removing components leads to an activation of different set of reactions, which in turn alters the LSS. Consequently, the gene expression levels are changed compared to the original (i.e., WT) state (**Figure 2B**). For the Snf1 pathway we removed the Snf1 protein component and this perturbation is referred as *snf1*Δ. For the *snf1*Δ model simulation of the predicted gene expression state only changed for the SUC2 genes in the "no glucose" conditions compared to the WT model. Perturbation of the Snf3/Rgt2 pathway was performed by removing Snf3 and Rgt2 from the model, this model is referred to as *rgt2*Δ*snf3*Δ. This perturbation showed a different gene expression state for both expression of the HXT and HXK gene groups than the WT-model. Finally, for the disruption of the cAMP-PKA signaling all components of the PKA-complex were removed from the system (Tpk1, Tpk2, Tpk3, and Bcy1). This perturbed model was designated *tpk1*Δ*tpk2*Δ*tpk3*Δ*bcy1*Δ and displayed a different predicted gene expression pattern for the PDS genes compared to the WT (**Figure 2B**). Although the *tpk1*Δ*tpk2*Δ*tpk3*Δ*bcy1*Δ showed continuously active PDS gene group it did not for the STRE gene group. This is because of a gap filling part that was added that caused Msn2 and Msn4 dephosphorylation in "glucose" conditions (**Table 1**). This dephosphorylation part caused inactivation of Msn2 and Msn4 even when the inactivation of Yak1 and Rim15 was disrupted in the *tpk1*Δ*tpk2*Δ*tpk3*Δ*bcy1*Δ model. To find out which crosstalk reaction can overcome the consequences of signaling disruption the effect of crosstalk on the altered gene expression patterns was analyzed. This was done by simulating all possible combinations of crosstalk 1-6 and 8 from **Table 2** in the "on" or "off" state. This resulted in 128 crosstalk combination vectors, which were used to activate crosstalk in the *snf1*Δ, the *rgt2*Δ*snf3*Δ, and the *tpk1*Δ*tpk2*Δ*tpk3*Δ*bcy1*Δ model. Simulations were only done for the environmental conditions that showed a different gene expression pattern, namely for *rgt2*Δ*snf3*Δ in "glucose", *tpk1*Δ*tpk2*Δ*tpk3*Δ*bcy1*Δ in "glucose" and *snf1*Δ in "no glucose" conditions. Each crosstalk reaction is active in half of the simulated crosstalk combinations. Every time a crosstalk reaction was active it was scored whether the predicted gene expression pattern behaved as the WT model or the perturbed system with all crosstalk reactions inactive (**Figure S3**). For *tpk1*Δ*tpk2*Δ*tpk3*Δ*bcy1*Δ in "glucose" and *snf1*Δ in "no glucose" and "glucose" conditions no combination of crosstalk reactions was able to overcome the effects of the perturbation (**Figures S3C–E**). Crosstalk 1 and 3 were shown to overcome the disruption effect of *rgt2*Δ*snf3*Δ in "glucose" conditions with every crosstalk combination they were active in. Crosstalk 1 and 3 are connections between the Snf1 and Rgt2/Snf3 pathway. If we simulated the *rgt2*Δ*snf3*Δ model with the connections between the Snf1 and Rgt2/Snf3 pathway included we were able to restore the WT gene expression pattern again (**Figure 2C** and **Figures S3A,B**). Considering a perturbed model, the crosstalk reactions that could restore the gene expression to the pattern predicted by the WT model may contribute to the signaling robustness of the yeast cell *in vivo*.

3.5. Addition of the TOR Pathway to the Model Shows Inter-connectivity Between Nitrogen and Glucose Signaling

The vector format rule-based modeling allows the model to be altered by addition of single components or even new pathways. Here, we added regulation by the nitrogen sensing TOR pathway (**Figure 4**). The TOR pathway regulation is interesting to consider since glucose sensing pathways Snf1 and PKA-cAMP and the nitrogen sensing pathways TOR have shown to be highly intertwined (Broach, 2012; Sanz et al., 2016). Therefore, we added the nitrogen sensing pathway to our model focusing on the Sch9 and PP2A downstream targets. The TOR pathway includes 15 proteins and one gap filler which controls the NCR genes

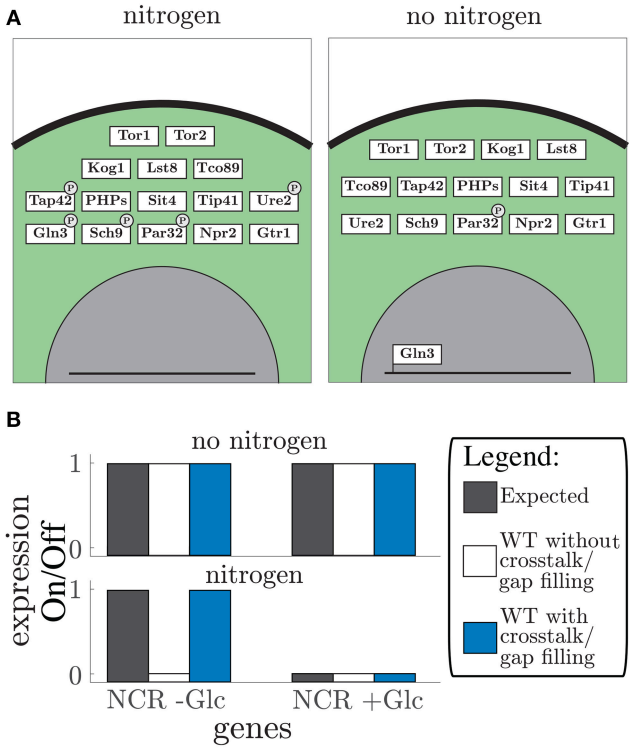


FIGURE 4 | TOR pathway: **(A)** the graphical representation of the simulated vector tables for the TOR pathway. Protein location is given by depicting the box in either the membrane (black line), cytosol (green area), or nucleus (gray area). Post-translational modifications such as phosphorylation are displayed by a black ellipse or circle on the protein box. The DNA bound status is give by the protein box being connected to the line (which displays DNA). **(B)** Expected gene expression pattern (black, left) compared to the predicted gene expression state from the model without (white, middle) and with addition of crosstalk reactions, and after the gap filling process (blue, right) for "no nitrogen" conditions (upper part) and "nitrogen" conditions (lower part) given for the NCR genes in addition with and without glucose (Glc).

(**Figure 4A**). The TOR complex 1 (TORC1) was handled as the second complex component in the model. This expanded the model to 67 components of which 57 proteins, 4 metabolites, and 6 unknown components (**Tables S6**, **S9**), adding another 10 conditions to the Boolean model (**Table S7**). Furthermore, it led to four additional crosstalk reactions (**Table S8**), which connected glucose and nitrogen signaling. These connections converge on two components: Rim15 in the PKA-cAMP pathway and Gln3 in the TOR pathway (Rødkær and Færgeman, 2014). The model shows the importance of Snf1 in glucose starvation, specifically, through NCR gene expression in addition to nitrogen starvation through mediation of Gln3 nuclear localization. Thereby expressing NCR genes, during glucose limitation, even in nitrogen rich conditions. This crosstalk reaction allows the cells to use amino acids as an alternative nitrogen and carbon source (Bertram et al., 2002). Note that even though TOR and Snf1 dependent phosphorylation of Gln3 have different phosphorylation sites (Bertram et al., 2002), they are treated equivalently in the model. In both single cases and in the hyper-phosphorylated state it corresponds to a phosphorylation status "1" in the state vector. These phosphorylation sites are considered equivalent because they both cause Gln5/mediator interaction. After adding crosstalk the model was capable of simulating the expected gene expression of the NCR genes Bertram et al. (2002) (**Figure 4B**). The gap filling process led to two unknown components (**Table S9**) that are responsible for dephosphorylation of Kog1 and Par32. These additional parts are only affecting the outcome when crosstalk is present. Remarkably, similar to the glucose signaling, information about protein phosphatases is missing. Along with the increased size of the model, nitrogen availability was included as an additional input, allowing twice as many possible combinations of nutrient inputs. By adding the TOR pathway to the model we showed that the model is easily extended by single components and whole pathways due to the simple structure and modularity. Furthermore, the importance of crosstalk in signaling pathways shows the inter-connectivity of glucose and nitrogen signaling.

4. DISCUSSION

To increase the information content of Boolean models from simple binary states, we assigned a vector to each component describing following features: localization, phosphorylation status, GDP/GTP exchange status, and DNA binding status (See section 2.1). Using this model, we found during the gap filling process that most lacking components are phosphatases, which indicates a lack of knowledge on phosphatases involved in nutrient sensing processes. The gap filling process also identified crosstalk from the PKA and Snf1 pathway to other pathways as a vital aspect to make the model switch between nutrient conditions. Model simulation of perturbed systems revealed that the crosstalk from the Snf1 pathway to the Rgt2/Snf3 pathway contributes to the robustness of this signaling network. The literature on nutrient sensing is quite extensive and this is a great resource to find mechanistic details on how the nutrient sensing network works. We set out to create a minimal system that can describe the RNA expression profile based on the input conditions. Most of the components and condition included in the model were shown in previous reports. However, for a few reactions different activation conditions were found, which are not mutually exclusive. Msn2 and Msn4 have been reported to be phosphorylated by Rim15, Yak1, and the PKA complex (Gorner et al., 2002; Lee et al., 2008, 2013). All these phosphorylation reactions have occurred in the active form of Msn2 and Msn4, although it is unclear which phosphorylation site(s) is/are deterministic for the function of Msn2 and Msn4. Since such reactions are closely related and appear almost simultaneously it is challenging distinguishing which reaction determines the occurrence of others, both computationally and experimentally. Such ambiguous mechanisms might result in multiple required conditions for a reaction to occur. All these conditions might not be representative *in vivo*, but do result in the same outcome as to be *in vivo* system. This is a limitation of modeling, since the model is only a representation of the knowledge we have of the system.

Since the knowledge gap in the literature did not allow us to create a model that could switch between nutrient conditions the gray areas needed to be filled in with a gap filling process. This network validation revealed that a common shortcoming on the knowledge of nutrient signaling pathways is how phosphate groups are removed from proteins, since the majority of the gaps in the model required addition of protein phosphatase reactions (**Table 1** and **Table S9**). This led us to identify protein phosphatases as major unknown components of the glucose signaling pathways. The addition of a component does not necessarily mean a protein function is missing, also degradation of a phosphorylated component has been identified as a efficient phosphatase system (Rowland et al., 2015). Most studies on signaling pathways focus on phosphorylation of proteins, but for a precise regulation dephosphorylation most also be tightly regulated. However, research has been biased toward phosphorylation event and therefore dephosphorylation of proteins has received much less attention (Castermans et al., 2012). High-throughput studies have identified around 40 different proteins as protein phosphatase in *S. cerevisiae* (Fiedler et al., 2009). This overabundance and the overlapping function of these protein phosphatases has made the identification of the exact function of these phosphatases a challenging task. To illustrate, three different protein phosphatases have shown to be responsible for Snf1 dephosphorylation, namely the protein phosphatase complex 1 Reg1-Glc7, Sir4, and Ptc1 (Ruiz et al., 2011, 2013; Zhang et al., 2011; Castermans et al., 2012). It remains unclear how the two latter are regulated by glucose and what their direct function is in nutrient signaling. Also, only recently has the Glc7-Reg1 protein phosphatase complex been identified as the Mig1 glucose-dependent phosphatase, however there is also a glucose independent dephosphorylation mechanism which is unknown (Shashkova et al., 2017). The lack of knowledge on protein phosphatase function is not restricted to nutrient signaling, and is absent in other pathways in yeast (Sacristan-Reviriego et al., 2015).

During the gap filling process we also found that known crosstalk reactions needed to be added to fill gaps (**Table 1**). Since these mainly included the PKA pathway it is suggested that this pathway has established crosstalk toward other pathways. These connections might be vital for the correct functioning of the carbon sensing network. This explains the observation that most glucose-responsive genes are regulated by a PKA-dependent pathway (Wang et al., 2004). Further, the inviability of the *tpk1*Δ*tpk2*Δ*tpk3*Δ triple mutant indicates the important role of the PKA complex in the cell (Pan and Heitman, 1999). This shows the importance of the PKA pathway as regulator of carbon availability and suggests the PKA pathway as a possible intervention point for drugs targeting nutrient sensing in cancer cells. This was confirmed with recent publications suggesting that intervention in the PKA signaling pathway might prove to be a effective strategy to eliminate cancer cells (Klutzny et al., 2018; Le et al., 2018; Wu et al., 2018).

The crosstalk analysis shown here suggests that the Snf1 pathway interaction with the Rgt2/Snf3 pathway contribute to the robustness of nutrient signaling, since crosstalk was able to overcome the perturbation of the Rgt2 and Snf3 components (**Figure 2C**). This shows the overlap between the Snf1 and the Rgt2/Snf3 pathway. Earlier study on downstream targets of these pathways, namely Mig1 and Mig2, have shown a considerable overlap of targeted promoters (Westholm et al., 2008). Also the connection from the Snf1 pathway to the TOR pathway maintains correct balance in metabolism and shows how interaction between signaling pathways maintain signaling robustness in the cell. This study, together with others, has shown that pathways are not linear and do not exist parallel next to each other. There is a significant crosstalk between pathways, which is essential for the functioning of nutrient signaling (Zaman et al., 2008). Classically a sensing pathway is viewed as a singular element. However, it seems that sensing pathways reside within a large regulatory network, which overlaps between the different pathways.

Further, addition of other signaling pathways to our model is straightforward, which we demonstrated with the inclusion of TOR pathway. This opens the path of adding sensing and signaling mechanisms for other essential nutrients such as macro-nutrients phosphate and sulfate or micro-nutrients like metal ions (Conrad et al., 2014; Bird, 2015; Qi et al., 2016; Samyn and Persson, 2016). Potentially this could contribute to the understanding of how the cell senses macro-nutrients, which provide the cell carbon, nitrogen, phosphorus and sulfur, or micro-nutrients, such as metal ions and vitamins. The realization of this complete model would increase the perception of how nutrient sensing systems achieve sensitive cellular gene expression reprogramming.

The Boolean modeling system created in this work is discrete, deterministic, and semi-quantitative. This is an oversimplification of real sensing networks, but this problem could be overcome using a probabilistic Boolean modeling approach. This approach would be able to add molecular and genetic noise to the model (Liang and Han, 2012; Zhu et al., 2014), which would allow the input and output of the model to be continuous instead of discrete. This added complexity would result in a model that can provide more mechanistic detail. However, this would require a more complicated computational setup, which might prove to be a trade-off toward the modularity.

Overall, in this work we have developed, simulated and validated a Boolean logic model describing the nutrient sensing network in yeast. The development and validation process revealed the importance of crosstalk from one pathway to other nutrient sensing pathways and showed that the unknown components in the glucose signaling pathway are mostly phosphatases. By studying the interactions within the nutrient sensing network this work contributes to the holistic understanding of nutrient sensing and shows the impact of crosstalk on network robustness and functioning.

AUTHOR CONTRIBUTIONS

NW and MC conceived the presented idea. NW and BS designed the model and the implementation. BS performed

the model simulations. NW and LÖ carried out the experiments. MC supervised the execution of the work. All authors discussed the results and contributed to the final manuscript.

FUNDING

This project is financially supported by Swedish Foundation for Strategic Research and Hasselblad Foundation.

ACKNOWLEDGMENTS

We want to thank Peter Dahl and Sviatlana Shashkova for providing us the yeast strains required for the experimental work.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphys.2018.01964/full#supplementary-material

REFERENCES

Abou-Jaoudé, W., Traynard, P., Monteiro, P. T., Saez-Rodriguez, J., Helikar, T., Thieffry, D., et al. (2016). Logical modeling and dynamical analysis of cellular networks. *Front. Genet.* 7:94. doi: 10.3389/fgene.2016.00094

Alberghina, L., and Westerhoff, H. V. (2005). *Systems Biology, Definitions and Perspectives*. Berlin; Heidelberg: Springer-Verlag.

Anderson, C. S., DeDiego, M. L., Topham, D. J., and Thakar, J. (2016). Boolean modeling of cellular and molecular pathways involved in influenza infection. *Comput. Math. Methods Med.* 2016:7686081. doi: 10.1155/2016/7686081

Barrett, L., Orlova, M., Maziarz, M., and Kuchin, S. (2012). Protein kinase A contributes to the negative control of Snf1 protein kinase in *Saccharomyces cerevisiae*. *Eukaryot. Cell* 11, 119–128. doi: 10.1128/EC.05061-11

Bertram, P. G., Choi, J. H., Carvalho, J., Chan, T.-F., Ai, W., and Zheng, X. F. S. (2002). Convergence of TOR-nitrogen and Snf1-glucose signaling pathways onto Gln3. *Mol. Cell. Biol.* 22, 1246–1252. doi: 10.1007/978-3-319-43589-37

Bird, A. J. (2015). Cellular sensing and transport of metal ions: implications in micronutrient homeostasis. *J. Nutr. Biochem.* 26, 1103–1115. doi: 10.1016/j.jnutbio.2015.08.002

Bontron, S., Jaquenoud, M., Vaga, S., Talarek, N., Bodenmiller, B., Aebersold, R., et al. (2013). Yeast endosulfines control entry into quiescence and chronological life span by inhibiting protein phosphatase 2A. *Cell Rep.* 3, 16–22. doi: 10.1016/j.celrep.2012.11.025

Bornholdt, S. (2008). Boolean network models of cellular regulation: prospects and limitations. *J. R. Soc. Interface* 5(Suppl. 1), S85–S94. doi: 10.1098/rsif.2008.0132.focus

Boutillier, P., Maasha, M., Li, X., Medina-Abarca, H. F., Krivine, J., Feret, J., et al. (2018). The Kappa platform for rule-based modeling. *Bioinformatics* 34, i583–i592. doi: 10.1093/bioinformatics/bty272

Broach, J. R. (1991). Ras-regulated signaling processes in *Saccharomyces cerevisiae*. *Curr. Opin. Genet. Dev.* 1, 370–377.

Broach, J. R. (2012). Nutritional control of growth and development in yeast. *Genetics* 192, 73–105. doi: 10.1534/genetics.111.135731

Cameroni, E., Hulo, N., Roosen, J., Winderickx, J., and De Virgilio, C. (2004). The novel yeast PAS kinase Rim 15 orchestrates G0-associated antioxidant defense mechanisms. *Cell Cycle* 3, 460–466. doi: 10.4161/cc.3.4.791

Castermans, D., Somers, I., Kriel, J., Louwet, W., Wera, S., et al. (2012). Glucose-induced posttranslational activation of protein phosphatases PP2A and PP1 in yeast. *Cell Res.* 22, 1058–1077. doi: 10.1038/cr.2012.20

Celenza, J. L., and Carlson, M. (1989). Mutational analysis of the *Saccharomyces cerevisiae* SNF1 protein kinase and evidence for functional interaction with the SNF4 protein. *Mol. Cell. Biol.* 9, 5034–5044.

Christensen, T. S., Oliveira, A. P., and Nielsen, J. (2009). Reconstruction and logical modeling of glucose repression signaling pathways in *Saccharomyces cerevisiae*. *BMC Syst. Biol.* 3:7. doi: 10.1186/1752-0509-3-7

Colombo, S., Ma, P., Cauwenberg, L., Winderickx, J., Crauwels, M., Teunissen, A., et al. (1998). Involvement of distinct G-proteins, Gpa2 and Ras, in glucose- and intracellular acidification-induced cAMP signalling in the yeast *Saccharomyces cerevisiae*. *EMBO J.* 17, 3326–3341. doi: 10.1093/emboj/17.12.3326

Colombo, S., Ronchetti, D., Thevelein, J. M., Winderickx, J., and Martegani, E. (2004). Activation state of the Ras2 protein and glucose-induced signaling in *Saccharomyces cerevisiae*. *J. Biol. Chem.* 279, 46715–46722. doi: 10.1074/jbc.M405136200

Conrad, M., Schothorst, J., Kankipati, H. N., Van Zeebroeck, G., Rubio-Texeira, M., and Thevelein, J. M. (2014). Nutrient sensing and signaling in the yeast *Saccharomyces cerevisiae*. *FEMS Microbiol. Rev.* 38, 254–299. doi: 10.1111/1574-6976.12065

De Vit, M. J., Waddle, J. A., and Johnston, M. (1997). Regulated nuclear translocation of the Mig1 glucose repressor. *Mol. Biol. Cell* 8, 1603–1618.

De Wever, V., Reiter, W., Ballarini, A., Ammerer, G., and Brocard, C. (2005). A dual role for PP1 in shaping the Msn2-dependent transcriptional response to glucose starvation. *EMBO J.* 24, 4115–4123. doi: 10.1038/sj.emboj.7600871

Düvel, K., Santhanam, A., Garrett, S., Schneper, L., and Broach, J. R. (2003). Multiple roles of Tap42 in mediating rapamycin-induced transcriptional changes in yeast. *Mol. Cell* 11, 1467–1478. doi: 10.1016/S1097-2765(03)00228-4

Endres, R. G. (2012). Signaling crosstalk: new insights require new vocabulary. *Biophys. J.* 103, 2241–2242. doi: 10.1016/j.bpj.2012.10.007

Fiedler, D., Braberg, H., Mehta, M., Chechik, G., Cagney, G., Mukherjee, P., et al. (2009). Functional organization of the *S. cerevisiae* phosphorylation network. *Cell* 136, 952–963. doi: 10.1016/j.cell.2008.12.039

Flick, K. M., Spielewoy, N., Kalashnikova, T. I., Guaderrama, M., Zhu, Q., Chang, H.-C., et al. (2003). Grr1-dependent inactivation of Mth1 mediates glucose-induced dissociation of Rgt1 from HXT gene promoters. *Mol. Biol. Cell* 14, 3230–3241. doi: 10.1091/mbc.E03-03-0135

Frolova, E., Johnston, M., and Majors, J. (1999). Binding of the glucose-dependent Mig1p repressor to the GAL1 and GAL4 promoters *in vivo*: regulationby glucose and chromatin structure. *Nucleic Acids Res.* 27, 1350–1358.

Gadura, N., Robinson, L. C., and Michels, C. A. (2006). Glc7-Reg1 phosphatase signals to Yck1,2 casein kinase 1 to regulate transport activity and glucose-induced inactivation of Saccharomyces maltose permease. *Genetics* 172, 1427–1439. doi: 10.1534/genetics.105.051698

Gancedo, J. M. (2008). The early steps of glucose signalling in yeast. *FEMS Microbiol. Rev.* 32, 673–704. doi: 10.1111/j.1574-6976.2008.00117.x

Georis, I., Feller, A., Tate, J. J., Cooper, T. G., and Dubois, E. (2009). Nitrogen catabolite repression-sensitive transcription as a readout of Tor pathway regulation: the genetic background, reporter gene and GATA factor assayed determine the outcomes. *Genetics* 181, 861–874. doi: 10.1534/genetics.108.099051

Görner, W., Durchschlag, E., Martinez-Pastor, M. T., Estruch, F., Ammerer, G., Hamilton, B., et al. (1998). Nuclear localization of the C2H2 zinc finger protein Msn2p is regulated by stress and protein kinase A activity. *Genes Dev.* 12, 586–597.

Görner, W., Durchschlag, E., Wolf, J., Brown, E. L., Ammerer, G., Ruis, H., et al. (2002). Acute glucose starvation activates the nuclear localization signal of a stress-specific yeast transcription factor. *EMBO J.* 21, 135–144. doi: 10.1093/emboj/21.1.135

Haney, S. A., and Broach, J. R. (1994). Cdc25p, the guanine nucleotide exchange factor for the Ras proteins of *Saccharomyces cerevisiae*, promotes exchange by stabilizing Ras in a nucleotide-free state. *J. Biol. Chem.* 269, 16541–16548.

Hlavacek, W. S., Faeder, J. R., Blinov, M. L., Posner, R. G., Hucka, M., and Fontana, W. (2006). Rules for modeling signal-transduction systems. *Sci. STKE* 2006:re6. doi: 10.1126/stke.3442006re6

Hong, S. P., Momcilovic, M., and Carlson, M. (2005). Function of mammalian LKB1 and Ca2+/calmodulin-dependent protein kinase kinase

alpha as Snf1-activating kinases in yeast. *J. Biol. Chem.* 280, 21804–21809. doi: 10.1074/jbc.M501887200

Horak, J. (2013). Regulations of sugar transporters: insights from yeast. *Current genetics*, 59(1-2):1–31. doi: 10.1007/s00294-013-0388-8

Hu, Z., Nehlin, J. O., Ronne, H., and Michels, C. A. (1995). MIG1-dependent and MIG1-independent glucose regulation of MAL gene expression in *Saccharomyces cerevisiae*. *Curr. Genet.* 28, 258–266.

Hubbard, E. J., Jiang, R., and Carlson, M. (1994). Dosage-dependent modulation of glucose repression by MSN3 (STD1) in *Saccharomyces cerevisiae*. *Mol. Cell. Biol.* 14, 1972–1978.

Hughes Hallett, J. E., Luo, X., and Capaldi, A. P. (2014). State transitions in the TORC1 signaling pathway and information processing in *Saccharomyces cerevisiae*. *Genetics* 198, 773–786. doi: 10.1534/genetics.114.168369

Jacinto, E., Guo, B., Arndt, K. T., Schmelzle, T., and Hall, M. N. (2001). TIP41 interacts with TAP42 and negatively regulates the TOR signaling pathway. *Mol. Cell* 8, 1017–1026. doi: 10.1016/S1097-2765(01)00386-0

Jouandot, D., Roy, A., and Kim, J. H. (2011). Functional dissection of the glucose signaling pathways that regulate the yeast glucose transporter gene (HXT) repressor Rgt1. *J. Cell. Biochem.* 112, 3268–3275. doi: 10.1002/jcb.23253

Kayikci, O., and Nielsen, J. (2015). Glucose repression in *Saccharomyces cerevisiae*. *FEMS Yeast Res.* 15:fov068. doi: 10.1093/femsyr/fov068

Kim, J. H., Brachet, V., Moriya, H., and Johnston, M. (2006). Integration of transcriptional and posttranslational regulation in a glucose signal transduction pathway in *Saccharomyces cerevisiae*. *Eukaryot. Cell* 5, 167–173. doi: 10.1128/EC.5.1.167-173.2006

Kim, J. H., Polish, J., and Johnston, M. (2003). Specificity and regulation of DNA binding by the yeast glucose transporter gene repressor Rgt1. *Mol. Cell. Biol.* 23, 5208–5216. doi: 10.1128/MCB.23.15.5208-5216.2003

Klutzny, S., Anurin, A., Nicke, B., Regan, J. L., Lange, M., Schulze, L., et al. (2018). PDE5 inhibition eliminates cancer stem cells via induction of PKA signaling. *Cell Death Dis.* 9, 192. doi: 10.1038/s41419-017-0202-5

Kruckeberg, A. L. (1996). The hexose transporter family of *Saccharomyces cerevisiae*. *Arch. Microbiol.* 166, 283–292.

Kuchin, S., Vyas, V. K., Kanter, E., Hong, S. P., and Carlson, M. (2003). Std1p (Msn3p) positively regulates the Snf1 kinase in *Saccharomyces cerevisiae*. *Genetics* 163, 507–514. Available online at: http://www.genetics.org/content/163/2/507

Kuroda, Y., Suzuki, N., and Kataoka, T. (1993). The effect of posttranslational modifications on the interaction of Ras2 with adenylyl cyclase. *Science* 259, 683–686.

Le, K., Steagall, W. K., Stylianou, M., Pacheco-Rodriguez, G., Darling, T. N., Vaughan, M., et al. (2018). Effect of beta-agonists on LAM progression and treatment. *Proc. Natl. Acad. Sci. U.S.A.* 115, E944–E953. doi: 10.1073/pnas.1719960115

Lee, P., Cho, B. R., Joo, H. S., and Hahn, J. S. (2008). Yeast Yak1 kinase, a bridge between PKA and stress-responsive transcription factors, Hsf1 and Msn2/Msn4. *Mol. Microbiol.* 70, 882–895. doi: 10.1111/j.1365-2958.2008.06450.x

Lee, P., Kim, M. S., Paik, S.-M., Choi, S.-H., Cho, B.-R., and Hahn, J.-S. (2013). Rim15-dependent activation of Hsf1 and Msn2/4 transcription factors by direct phosphorylation in *Saccharomyces cerevisiae*. *FEBS Lett.* 587, 3648–3655. doi: 10.1016/j.febslet.2013.10.004

Lee, P., Paik, S. M., Shin, C. S., Huh, W. K., and Hahn, J. S. (2011). Regulation of yeast Yak1 kinase by PKA and autophosphorylation-dependent 14-3-3 binding. *Mol. Microbiol.* 79, 633–646. doi: 10.1111/j.1365-2958.2010.07471.x

Liang, J., and Han, J. (2012). Stochastic Boolean networks: an efficient approach to modeling gene regulatory networks. *BMC Syst. Biol.* 6:113. doi: 10.1186/1752-0509-6-113

Loewith, R., and Hall, M. N. (2011). Target of rapamycin (TOR) in nutrient signaling and growth control. *Genetics* 189, 1177–1201. doi: 10.1534/genetics.111.133363

Lubitz, T., Welkenhuysen, N., Shashkova, S., Bendrioua, L., Hohmann, S., Klipp, E., et al. (2015). Network reconstruction and validation of the Snf1/AMPK pathway in baker's yeast based on a comprehensive literature review. *npj Syst. Biol. Appl.* 1:15007. doi: 10.1038/npjsba.2015.7

Lutfiyya, L. L., Iyer, V. R., DeRisi, J., DeVit, M. J., Brown, P. O., and Johnston, M. (1998). Characterization of three related glucose repressors and genes they regulate in *Saccharomyces cerevisiae*. *Genetics* 150, 1377–1391.

Ma, H., Han, B. K., Guaderrama, M., Aslanian, A., Yates, J. R., Hunter, T., et al. (2014). Psy2 targets the PP4 family phosphatase Pph3 to dephosphorylate Mth1 and repress glucose transporter gene expression. *Mol. Cell. Biol.* 34, 452–463. doi: 10.1128/MCB.00279-13

Martinez-Pastor, M. T., Marchler, G., Schuller, C., Marchler-Bauer, A., Ruis, H., and Estruch, F. (1996). The *Saccharomyces cerevisiae* zinc finger proteins Msn2p and Msn4p are required for transcriptional induction through the stress response element (STRE). *EMBO J.* 15, 2227–2235.

Mitts, M. R., Bradshaw-Rouse, J., and Heideman, W. (1991). Interactions between adenylate cyclase and the yeast GTPase-activating protein IRA1. *Mol. Cell. Biol.* 11, 4591–4598.

Moriya, H., and Johnston, M. (2004). Glucose sensing and signaling in *Saccharomyces cerevisiae* through the Rgt2 glucose sensor and casein kinase I. *Proc. Natl. Acad. Sci. U.S.A.* 101, 1572–1577. doi: 10.1073/pnas.0305901101

Mosley, A. L., Lakshmanan, J., Aryal, B. K., and Ozcan, S. (2003). Glucose-mediated phosphorylation converts the transcription factor Rgt1 from a repressor to an activator. *J. Biol. Chem.* 278, 10322–10327. doi: 10.1074/jbc.M212802200

Nicastro, R., Tripodi, F., Gaggini, M., Castoldi, A., Reghellin, V., Nonnis, S., et al. (2015). Snf1 phosphorylates adenylate cyclase and negatively regulates protein kinase A-dependent transcription in *Saccharomyces cerevisiae*. *J. Biol. Chem.* 290, 24715–24726. doi: 10.1074/jbc.M115.658005

Orzechowski Westholm, J., Tronnersjö, S., Nordberg, N., Olsson, I., Komorowski, J., and Ronne, H. (2012). Gis1 and Rph1 regulate glycerol and acetate metabolism in glucose depleted yeast cells. *PLoS ONE* 7:e31577. doi: 10.1371/journal.pone.0031577

Ozcan, S., Dover, J., Rosenwald, A. G., Wölfl, S., and Johnston, M. (1996). Two glucose transporters in *Saccharomyces cerevisiae* are glucose sensors that generate a signal for induction of gene expression. *Proc. Natl. Acad. Sci.U.S.A.* 93, 12428–12432.

Ozcan, S., and Johnston, M. (1995). Three different regulatory mechanisms enable yeast hexose transporter (HXT) genes to be induced by different levels of glucose. *Mol. Cell. Biol.* 15, 1564–1572.

Palomino, A., Herrero, P., and Moreno, F. (2005). Rgt1, a glucose sensing transcription factor, is required for transcriptional repression of the HXK2 gene in *Saccharomyces cerevisiae*. *Biochem. J.* 388(Pt 2), 697–703. doi: 10.1042/BJ20050160

Pan, X., and Heitman, J. (1999). Cyclic AMP-dependent protein kinase regulates pseudohyphal differentiation in *Saccharomyces cerevisiae*. *Mol. Cell. Biol.* 19, 4874–4887.

Papin, J. A., Hunter, T., Palsson, B. O., and Subramaniam, S. (2005). Reconstruction of cellular signalling networks and analysis of their properties. *Nat. Rev. Mol. Cell Biol.* 6, 99–111. doi: 10.1038/nrm1570

Pasula, S., Jouandot, D., and Kim, J.-H. (2007). Biochemical evidence for glucose-independent induction of HXT expression in *Saccharomyces cerevisiae*. *FEBS Lett.* 581, 3230–3234. doi: 10.1016/j.febslet.2007.06.013

Pedruzzi, I., Bürckert, N., Egger, P., and De Virgilio, C. (2000). *Saccharomyces cerevisiae* Ras/cAMP pathway controls post-diauxic shift element-dependent transcription through the zinc finger protein Gis1. *EMBO J.* 19, 2569–2579. doi: 10.1093/emboj/19.11.2569

Peeters, T., Versele, M., and Thevelein, J. M. (2007). Directly from Galpha to protein kinase A: the kelch repeat protein bypass of adenylate cyclase. *Trends Biochem. Sci.* 32, 547–554. doi: 10.1016/j.tibs.2007.09.011

Qi, W., Baldwin, S. A., Muench, S. P., and Baker, A. (2016). Pi sensing and signalling: from prokaryotic to eukaryotic cells. *Biochem. Soc. Trans.* 44, 766–773. doi: 10.1042/BST20160026

Rødkær, S. V., and Færgeman, N. J. (2014). Glucose- and nitrogen sensing and regulatory mechanisms in *Saccharomyces cerevisiae*. *FEMS Yeast Res.* 14, 683–696. doi: 10.1111/1567-1364.12157

Roosen, J., Engelen, K., Marchal, K., Mathys, J., Griffioen, G., Cameroni, E., et al. (2005). PKA and Sch9 control a molecular switch important for the proper adaptation to nutrient availability. *Mol. Microbiol.* 55, 862–880. doi: 10.1111/j.1365-2958.2004.04429.x

Rother, M., Münzner, U., Thieme, S., and Krantz, M. (2013). Information content and scalability in signal transduction network reconstruction formats. *Mol. Biosyst.* 9, 1993–2004. doi: 10.1039/c3mb00005b

Rowland, M. A., and Deeds, E. J. (2014). Crosstalk and the evolution of specificity in two-component signaling. *Proc. Natl. Acad. Sci.U.S.A.* 111, 5550–5555. doi: 10.1073/pnas.1317178111

Rowland, M. A., Fontana, W., and Deeds, E. J. (2012). Crosstalk and competition in signaling networks. *Biophys. J.* 103, 2389–2398. doi: 10.1016/j.bpj.2012.10.006

Rowland, M. A., Harrison, B., and Deeds, E. J. (2015). Phosphatase specificity and pathway insulation in signaling networks. *Biophys. J.* 108, 986–996. doi: 10.1016/j.bpj.2014.12.011

Roy, A., Shin, Y. J., Cho, K. H., and Kim, J.-H. (2013). Mth1 regulates the interaction between the Rgt1 repressor and the Ssn6-Tup1 corepressor complex by modulating PKA-dependent phosphorylation of Rgt1. *Mol. Biol. Cell* 24, 1493–1503. doi: 10.1091/mbc.E13-01-0047

Rubenstein, E. M., McCartney, R. R., Zhang, C., Shokat, K. M., Shirra, M. K., Arndt, K. M., et al. (2008). Access denied: Snf1 activation loop phosphorylation is controlled by availability of the phosphorylated threonine 210 to the PP1 phosphatase. *J. Biol. Chem.* 283, 222–230. doi: 10.1074/jbc.M707957200

Ruiz, A., Xu, X., and Carlson, M. (2011). Roles of two protein phosphatases, Reg1-Glc7 and Sit4, and glycogen synthesis in regulation of SNF1 protein kinase. *Proc. Natl. Acad. Sci.U.S.A.* 108, 6349–6354. doi: 10.1073/pnas.1102758108

Ruiz, A., Xu, X., and Carlson, M. (2013). Ptc1 protein phosphatase 2C contributes to glucose regulation of SNF1/AMP-activated protein kinase (AMPK) in *Saccharomyces cerevisiae*. *J. Biol. Chem.* 288, 31052–31058. doi: 10.1074/jbc.M113.503763

Sacristan-Reviriego, A., Martin, H., and Martín, M. (2015). Identification of putative negative regulators of yeast signaling through a screening for protein phosphatases acting on cell wall integrity and mating MAPK pathways. *Fungal Genet. Biol.* 77, 1–11. doi: 10.1016/j.fgb.2015.02.011

Samyn, D. R., and Persson, B. L. (2016). Inorganic phosphate and sulfate transport in *S. cerevisiae*. *Adv. Exp. Med. Biol.* 892, 253–269. doi: 10.1007/978-3-319-25304-610

Sanz, P., Viana, R., and Garcia-Gimeno, M. A. (2016). AMPK in yeast: the SNF1 (Sucrose Non-fermenting 1) protein kinase complex. *EXS* 107, 353–374. doi: 10.1007/978-3-319-43589-314

Schlatter, R., Schmich, K., Avalos Vizcarra, I., Scheurich, P., Sauter, T., Borner, C., et al. (2009). ON/OFF and beyond–a boolean model of apoptosis. *PLoS Comput. Biol.* 5:e1000595. doi: 10.1371/journal.pcbi.1000595

Schmidt, M. C., and McCartney, R. R. (2000). beta-subunits of Snf1 kinase are required for kinase function and substrate definition. *EMBO J.* 19, 4936–4943. doi: 10.1093/emboj/19.18.4936

Schmitt, A. P., and McEntee, K. (1996). Msn2p, a zinc finger DNA-binding protein, is the transcriptional activator of the multistress response in *Saccharomyces cerevisiae*. *Proc. Natl. Acad. Sci.U.S.A.* 93, 5777–5782.

Shashkova, S., Welkenhuysen, N., and Hohmann, S. (2015). Molecular communication: crosstalk between the Snf1 and other signaling pathways. *FEMS Yeast Res.* 15, 1–10. doi: 10.1093/femsyr/fov026

Shashkova, S., Wollman, A. J. M., Leake, M. C., and Hohmann, S. (2017). The yeast Mig1 transcriptional repressor is dephosphorylated by glucose-dependent and -independent mechanisms. *FEMS Microbiol. Lett.* 364, 1–9. doi: 10.1093/femsle/fnx133

Singh, A., Nascimento, J. M., Kowar, S., Busch, H., and Boerries, M. (2012). Boolean approach to signalling pathway modelling in HGF-induced keratinocyte migration. *Bioinformatics* 28, i495–i501. doi: 10.1093/bioinformatics/bts410

Swinnen, E., Wanke, V., Roosen, J., Smets, B., Dubouloz, F., Pedruzzi, I., et al. (2006). Rim15 and the crossroads of nutrient signalling pathways in *Saccharomyces cerevisiae*. *Cell Division* 1:3. doi: 10.1186/1747-1028-1-3

Tate, J. J., Georis, I., Dubois, E., and Cooper, T. G. (2010). Distinct phosphatase requirements and GATA factor responses to nitrogen catabolite repression and rapamycin treatment in *Saccharomyces cerevisiae*. *J. Biol. Chem.* 285, 17880–17895. doi: 10.1074/jbc.M109.085712

Tomás-Cobos, L. and Sanz, P. (2002). Active Snf1 protein kinase inhibits expression of the *Saccharomyces cerevisiae* HXT1 glucose transporter gene. *Biochem. J.* 368(Pt 2), 657–663. doi: 10.1042/BJ20020984

Treitel, M. A., and Carlson, M. (1995). Repression by SSN6-TUP1 is directed by MIG1, a repressor/activator protein. *Proc. Natl. Acad. Sci.U.S.A.* 92, 3132–3136.

Vert, G., and Chory, J. (2011). Crosstalk in cellular signaling: background noise or the real thing? *Dev. Cell* 21, 985–991. doi: 10.1016/j.devcel.2011.11.006

Wang, R.-S., Saadatpour, A., and Albert, R. (2012). Boolean modeling in systems biology: an overview of methodology and applications. *Phys. Biol.* 9:55001. doi: 10.1088/1478-3975/9/5/055001

Wang, Y., Pierce, M., Schneper, L., Guldal, C. G., Zhang, X., Tavazoie, S., et al. (2004). Ras and Gpa2 mediate one branch of a redundant glucose signaling pathway in yeast. *PLoS Biol.* 2:E128. doi: 10.1371/journal.pbio.0020128

Westholm, J. O., Nordberg, N., Murén, E., Ameur, A., Komorowski, J., and Ronne, H. (2008). Combinatorial control of gene expression by the three yeast repressors Mig1, Mig2 and Mig3. *BMC Genomics* 9:601. doi: 10.1186/1471-2164-9-601

Wu, J., Gao, F., Xu, T., Deng, X., Wang, C., Yang, X., et al. (2018). miR-503 suppresses the proliferation and metastasis of esophageal squamous cell carcinoma by triggering autophagy via PKA/mTOR signaling. *Int. J. Oncol.* 52, 1427–1442. doi: 10.3892/ijo.2018.4320

Zaman, S., Lippman, S. I., Zhao, X., and Broach, J. R. (2008). How Saccharomyces responds to nutrients. *Annu. Rev. Genet.* 42, 27–81. doi: 10.1146/annurev.genet.41.110306.130206

Zhang, Y., McCartney, R. R., Chandrashekarappa, D. G., Mangat, S., and Schmidt, M. C. (2011). Reg1 protein regulates phosphorylation of all three Snf1 isoforms but preferentially associates with the Gal83 isoform. *Eukaryot. Cell* 10, 1628–1636. doi: 10.1128/EC.05176-11

Zhu, P., Liang, J., and Han, J. (2014). Gene perturbation and intervention in context-sensitive stochastic Boolean networks. *BMC Syst. Biol.* 8:60. doi: 10.1186/1752-0509-8-60

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Physiology

ORIGINAL RESEARCH
published: 24 January 2019
doi: 10.3389/fphys.2018.01965

Personalization of Logical Models With Multi-Omics Data Allows Clinical Stratification of Patients

Jonas Béal, Arnau Montagud, Pauline Traynard, Emmanuel Barillot and Laurence Calzone**

Institut Curie, PSL Research University, Mines Paris Tech, Inserm, U900, Paris, France

Logical models of cancer pathways are typically built by mining the literature for relevant experimental observations. They are usually generic as they apply for large cohorts of individuals. As a consequence, they generally do not capture the heterogeneity of patient tumors and their therapeutic responses. We present here a novel framework, referred to as PROFILE, to tailor logical models to a particular biological sample such as a patient tumor. This methodology permits to compare the model simulations to individual clinical data, i.e., survival time. Our approach focuses on integrating mutation data, copy number alterations (CNA), and expression data (transcriptomics or proteomics) to logical models. These data need first to be either binarized or set between 0 and 1, and can then be incorporated in the logical model by modifying the activity of the node, the initial conditions or the state transition rates. The use of MaBoSS, a tool based on Monte-Carlo kinetic algorithm to perform stochastic simulations on logical models results in model state probabilities, and allows for a semi-quantitative study of the model phenotypes and perturbations. As a proof of concept, we use a published generic model of cancer signaling pathways and molecular data from METABRIC breast cancer patients. For this example, we test several combinations of data incorporation and discuss that, with these data, the most comprehensive patient-specific cancer models are obtained by modifying the nodes' activity of the model with mutations, in combination or not with CNA data, and altering the transition rates with RNA expression. We conclude that these model simulations show good correlation with clinical data such as patients' Nottingham prognostic index (NPI) subgrouping and survival time. We observe that two highly relevant cancer phenotypes derived from personalized models, *Proliferation* and *Apoptosis*, are biologically consistent prognostic factors: patients with both high proliferation and low apoptosis have the worst survival rate, and conversely. Our approach aims to combine the mechanistic insights of logical modeling with multi-omics data integration to provide patient-relevant models. This work leads to the use of logical modeling for precision medicine and will eventually facilitate the choice of patient-specific drug treatments by physicians.

Keywords: logical models, personalized mechanistic models, personalized medicine, breast cancer, data discretization, stochastic simulations

OPEN ACCESS

Edited by:
Matteo Barberis,
University of Surrey, United Kingdom

Reviewed by:
Olaf Wolkenhauer,
University of Rostock, Germany
Maximino Aldana,
National Autonomous University of Mexico, Mexico

****Correspondence:***
Emmanuel Barillot
emmanuel.barillot@curie.fr
Laurence Calzone
laurence.calzone@curie.fr

Specialty section:
This article was submitted to Systems Biology, a section of the journal Frontiers in Physiology

Received: *01 March 2018*
Accepted: *31 December 2018*
Published: *24 January 2019*

Citation:
Béal J, Montagud A, Traynard P, Barillot E and Calzone L (2019) Personalization of Logical Models With Multi-Omics Data Allows Clinical Stratification of Patients. Front. Physiol. 9:1965. doi: 10.3389/fphys.2018.01965

1. INTRODUCTION

Molecular profiling of patient samples is now becoming clinical routine in diseases like cancer, where it has shown therapeutic utility. Typically, tumor DNA or RNA are sequenced, and if an oncogene mutation is found, then it opens the opportunity to treat the patient with a targeted inhibitory drug which counteracts the mutated oncoprotein effect. Nevertheless, this strategy has often limited impact, because the tumor will eventually activate compensatory pathways or acquire novel mutations and escape the treatment. To anticipate drug resistance and optimize treatments, a better understanding of the regulatory network dynamics is needed. As a consequence, mathematical modeling has been increasingly used to formally describe the dynamics of regulatory networks representing the signaling pathways that are frequently altered in cancers. Many of these signaling pathways, e.g., apoptosis, mTOR pathway, RTK signaling, or DNA repair pathways, are shared among diverse cancers and contain common mutations or gene alterations. The translation of the networks recapitulating these pathways into mathematical models can be done using different formalisms. Over the past decades, numerous uses of logical modeling have shown that this framework is able to characterize the main dynamical properties of complex biological regulatory networks (Faure et al., 2006; Abou-Jaoudé et al., 2011; Grieco et al., 2013), as well as to predict the behavior of molecular networks affected in human diseases (Fumiã and Martins, 2013; Arshad and Datta, 2017).

However, these models usually describe general processes and tend to be generic, missing patients' specificities and possible patient-tailored interventions. To avoid the relapse that follows many treatments, these models need to be adjusted to each individual patient, capitalizing on omics profile of the patient tumor. Some work has been done on trying to contextualize these models to perturbation data (often (phospho-)proteomics data) (Saez-Rodriguez et al., 2009; Rodriguez et al., 2015; Dorier et al., 2016) but it remains difficult to apply these methods to patient data (typically genome and transcriptome) and get clinical insight. Additionally, some network-based methods have been investigated for patient stratification, using network propagation with somatic mutations (Hofree et al., 2013) or applying propagation of gene expression data on KEGG pathways coupled with mutation information (Hidalgo et al., 2017).

Our PROFILE (**PeR**sonalization **OF** log**I**ca**L** Mod**El**s) approach aims to combine the mechanistic insights of logical modeling with multi-omics data integration to provide patient-relevant models (**Figure 1**). The generic logical model can be any model in standard format, automatically translated into a format specific to MaBoSS (**Ma**rkovian **Bo**olean **S**tochastic **S**imulator), a tool that simulates continuous time Markov processes on Boolean networks (Stoll et al., 2012, 2017). The biological data are extracted from existing repositories or from private sources into a data frame per data type. The merging of these two inputs provides a personalized logical model per patient. Therefore, we define a personalization of a logical model as a specification of a generic logical model using available patient data. We present here a framework to tailor a logical model to patient-specific multi-omics data, thereby personalizing these generic models to particular patients or sets of patients with the goal to treat these patients in a personalized manner. We also show how to best use mutation, copy number and transcriptome patient data for model personalization. To illustrate the method, we gathered 2,509 breast cancer data genomic profiles from METABRIC project, including somatic mutations, copy number alterations, and gene expression (Curtis et al., 2012; Pereira et al., 2016), and integrated the data on a published logical model of generic cancer pathways (Fumiã and Martins, 2013) using MaBoSS. Lastly, we show evidence that our patient-specific models can be used to stratify patients by groups and by survival data.

We conclude that this framework allows us to provide models that can capture detailed descriptions of patient data, paving the way to modeling patient response to many potential targeted treatments or combination of treatments, and helping the clinical oncologists to choose the best option for personalized treatment (**Figure 1**). The framework can be used on any logical model, available in databases such as Cell Collective (https://cellcollective.org), and with any set of patient data, and thus used by non-experts in modeling.

It is freely available on GitHub (https://github.com/sysbio-curie/PROFILE) and is distributed open source under the BSD 3-clause license.

2. MATERIALS AND METHODS

2.1. Logical Modeling

2.1.1. Principles

Although continuous mathematical modeling based on chemical kinetics has been widely used to study cellular biochemistry dynamics (e.g., ordinary differential equations) (Novák and Tyson, 2004; Fey et al., 2015), this formalism faces limits for modeling large-scale signaling networks, due to the difficulty of estimating kinetic parameter values. In contrast, the logical modeling formalism represents a convenient mean of abstraction, where the causal relationships between proteins (or genes) are encoded with logical statements and dynamical behaviors are represented by transitions between discrete states of the system. The logical formalism is flexible, requires in principle no quantitative information, and, hence, can be applied to large networks combining multiple pathways. It can also provide a qualitative understanding of molecular systems lacking mechanistic detailed information. A brief summary of the main features of logical modeling is provided hereunder and a more detailed primer can be found in **Supplementary Material**. For more in-depth reviews on logical models, their construction and analyses, we refer the reader to several sources (Saadatpour and Albert, 2013; Le Novère, 2015; Abou-Jaoudé et al., 2016).

A logical model is based on a regulatory graph, where each node represents a component (e.g., a protein, gene, complex, process, etc.), and is associated with discrete levels of activity (0, 1, or more when justified) as represented in **Figure 2A**. Each edge corresponds to a regulatory interaction between the source and target nodes, and is represented by a positive or negative influence, depending on the type of regulation. Logical rules (or functions) are assigned to each node of the network. These rules

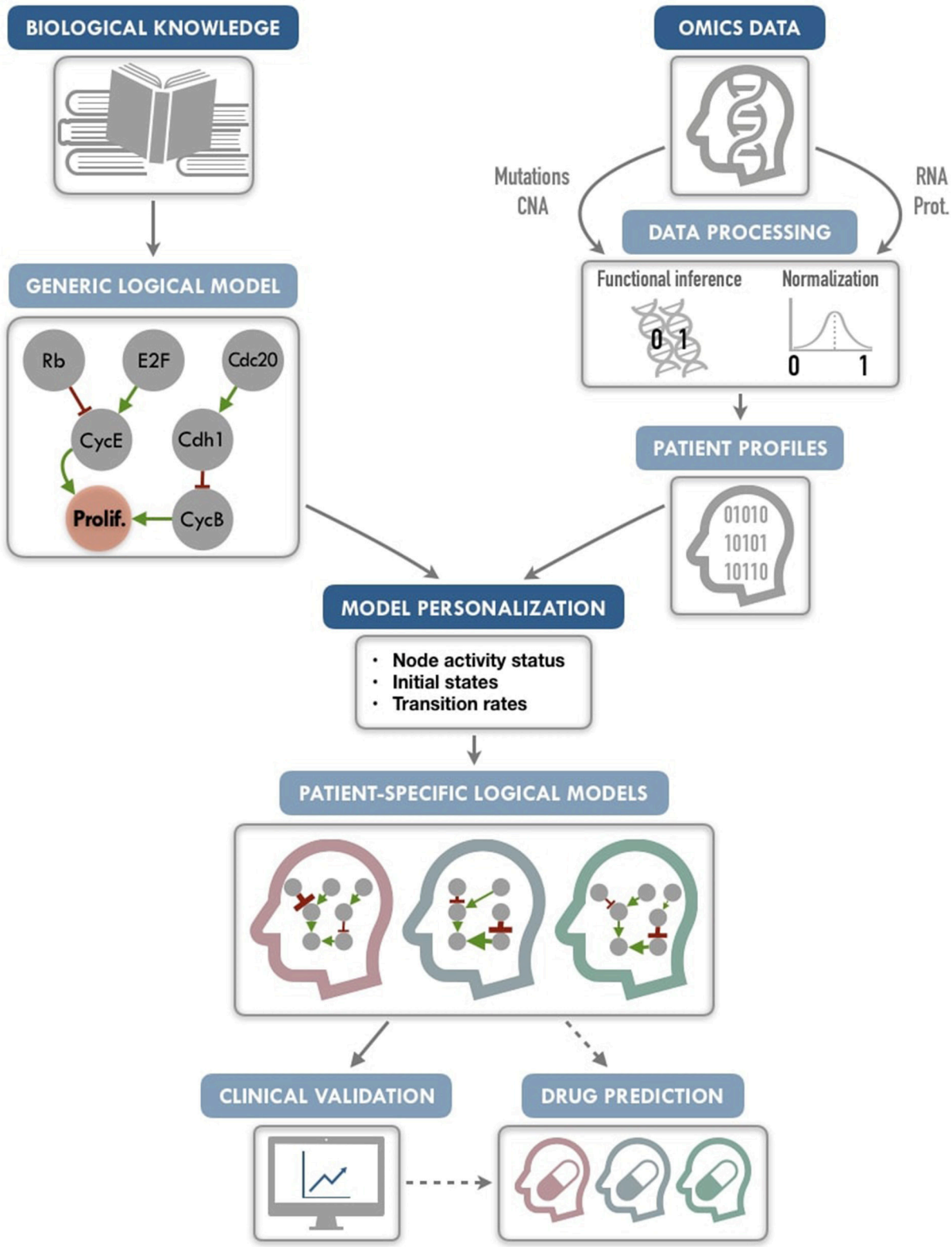


FIGURE 1 | PROFILE methodology for personalization of logical models. On the one hand (upper left), a generic logical model, in a MaBoSS format (a BND file for model description with logical rules and a CFG file for definition of the simulation parameters), is selected to serve as the starting-point. Note that any SBML qual

(Continued)

FIGURE 1 | model can be easily translated into a MaBoSS format. The parameters related to the nodes (initial states and transition rates) are chosen to be generic in the initial CFG file. On the other hand (upper right), omics data are gathered (e.g., genome and transcriptome) as data frames, and processed through functional inference methods (for already discrete genome data) or binarization/normalization (for continuous expression data). The resulting patient profiles are used to perform model personalization, i.e., adapt the generic model with patient data. The merging of the generic model with the patient profiles creates a personalized MaBoSS model with an unchanged BND file and a CFG file per patient. Then, clinical relevance of these patient-specific models can be assessed before providing original and personalized therapeutic strategies and drug predictions.

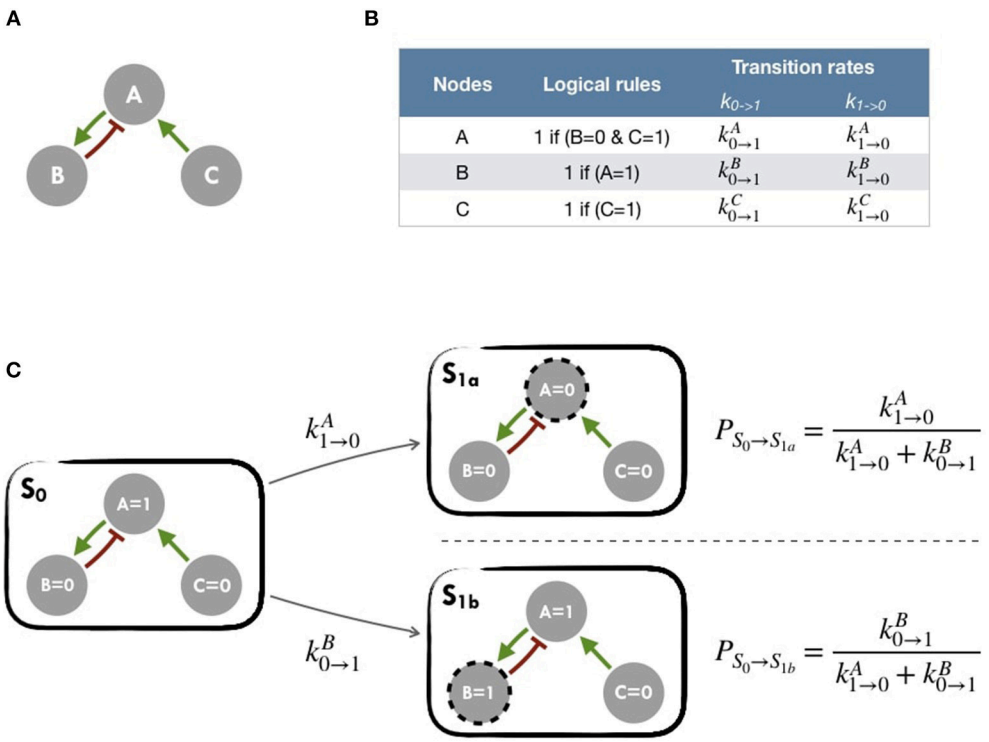


FIGURE 2 | Main principles of MaBoSS simulation framework and Gillespie algorithm **(A)** Toy model with B and C regulated by A. **(B)** In the first column, logical rules of the logical model: as an input A remains in its initial state; presence of A triggers B activation and C inhibition. In the second column, MaBoSS activation and inactivation transition rates are defined for each possible transition **(C)** In an asynchronous update scheme, starting from S_0 state, there are two possible following states S_{1a} and S_{1b} with their corresponding probabilities.

connect input nodes with logical operators AND (&), OR (|) and NOT (!), or a combination of these operators (**Figure 2B**). An example of a toy model can be found in **Figure 2A** and **Figure S1**.

The resulting dynamics can be represented in terms of a second type of graphs, the state transition graph (STG), where the nodes account for the states of the system, referred to as the model states (**Figure 2C**). The model states correspond to vectors of the nodes' activity, and the edges to the possible state transitions from one model state to another. When concurrent variable changes are enabled at a given state, the resulting state transition depends on the chosen updating assumption. Numerous studies use the simple fully synchronous strategy where all variables are updated through a unique transition (Weinstein et al., 2017). This assumption leads to relatively simple STG and deterministic dynamics (Helikar et al., 2008; Fumiã and Martins, 2013; Cho et al., 2016). However, the synchronous updating assumption may lead to spurious cyclic attractors. The asynchronous updating strategy considers separately all possible transitions and therefore provides alternative dynamics in the absence of kinetic data. The resulting dynamics have a branching structure that complicates its evaluation. An example of such graphs can be found in **Figure 2C** or **Figure S2** for an asynchronous graph and **Figure S3** for a synchronous graph.

In this work, asynchronous dynamics with stochastic simulations have been considered.

More details of logical models and their uses can be found in other works such as Abou-Jaoudé et al. (2016) and Chaouiya et al. (2012).

2.1.2. Simulations With MaBoSS

MaBoSS software is applied to obtain probabilities for each of the model states of the system using continuous time Markov chain simulations on the Boolean network (Stoll et al., 2012, 2017). Its principles are summarized in **Figure 2** and in **Figure S5** for a more comprehensive version. MaBoSS uses a specific language for associating transition rates, $k_{0\to1}$ (or k_{up}) and $k_{1\to0}$ (or k_{down}), to each node (**Figure 2B**), enabling to account for different time

scales of the processes described by the model. Given some initial conditions (i.e., either 0 or 1 state for each node), MaBoSS applies Monte-Carlo kinetic algorithm (or Gillespie algorithm) to the network.

This algorithm provides a stochastic way to choose a specific transition among several possible ones (**Figure 2C**), to perform asynchronous updates and finally to infer a corresponding time for this transition (**Figure S5D**). Thus, by concatenating stochastic updates, MaBoSS computation results in one stochastic trajectory as a function of time. The transition rates can be understood as probabilities in order to determine the actual transition. For our simulations, unless otherwise specified, all transition states were initially assigned to 1. Since MaBoSS computes stochastic trajectories, it is relevant to generate a population of stochastic trajectories to gain insight into the average behavior over the asynchronous STG.

The aggregation of stochastic trajectories can also be interpreted as a description of an heterogeneous population. Since several trajectories are simulated, initial values of each node can be defined with a continuous value between 0 and 1 representing the probability for the node to be defined to 1 for each new trajectory. For instance, a node with a 0.6 initial condition will be set to 1 in 60% of simulated trajectories and to 0 in 40% of them.

Two files are needed to run MaBoSS: a model file (BND) where the nodes of the model and their logical rules are listed and a configuration file (CFG) where initial states, transition rates and other parameters of the simulation are specified.

In the present work, all simulations were performed with MaBoSS and the focus has been set on the probabilities of nodes and phenotypes at the asymptotic state. Indeed, asymptotic states are more closely related to logical model attractors than transient dynamics. They are therefore less dependent on updating stochasticity and are more meaningful biologically (Huang et al., 2009).

Only 1,000 stochastic trajectories were computed in all simulations since it appeared as a sufficient number to obtain a median standard deviation below 0.01 (see **Figure S9**). For any study using MaBoSS, to insure that the state space is well explored, it is advised to start with a higher number of trajectories at first and reduce it when the median deviation is below a reasonable threshold.

Examples of MaBoSS applied to biological questions can be found in Calzone et al. (2010); Cohen et al. (2015); Remy et al. (2015); or Montagud et al. (2017). Any logical model in SBML qual format (Chaouiya et al., 2013) can be exported from GINsim (Chaouiya et al., 2012) into MaBoSS format, allowing the use of any logical model from databases for the PROFILE framework.

2.1.3. Generic Logical Model of Cancer Pathways

A published Boolean network model was used to illustrate our PROFILE methodology (Fumiã and Martins, 2013). It is based on a regulatory network summarizing several key players and pathways involved in cancer mechanisms: RTKs, PI3K/AKT, WNT/β-catenin, TGF-β/Smads, Rb, HIF-1, p53 and ATM/ATR. An input node *Acidosis* and an output node *Proliferation* used as a read-out were added to ease the analysis. Based on the model's logical rules from Fumiã and Martins (2013), *Proliferation* node is activated by any of the cyclins (*CyclinA*, *CyclinB*, *CyclinD*, and *CyclinE*) and is, thus, an indicator of cyclin activity as an abstraction of the cell cycle behavior. This is a simplification of cell cycle, and if readers would like to go beyond this abstraction, a detailed study on the dynamics of a mammalian cell cycle that takes into account cyclins and cyclin-dependent kinases can be found in Gérard and Goldbeter (2016). The generic model of Fumiã and Martins (2013) contains 98 nodes and 254 edges, and can be visually inspected in **Figure S6**. It is available in MaBoSS format in our GitHub repository: (https://github.com/sysbio-curie/PROFILE/tree/master/Models/Fumia2013).

2.2. Generation of Patient Profiles From Multi-Omics Datasets

2.2.1. TCGA and METABRIC Data

Patient data from METABRIC (Curtis et al., 2012; Pereira et al., 2016) with RNA expression data (n = 1,904), mutation profiles (n = 2,509), CNA (n = 2,173) and clinical data (n = 1,980) were gathered. Missing values were considered on an personalization-specific basis: if the personalization method used mutation profiles and RNA data, only the patients with data of these types were considered. More details on the abundance of data types' samples can be found in **Figure S11A**.

Breast cancer patient data from TCGA (Cancer Genome Atlas Network, 2012; Ciriello et al., 2015) with RNA expression data (n = 816), mutation profiles (n = 817), CNA (n = 816) and clinical data (n = 817) were also gathered. For TCGA RNA expression data, data from healthy samples are available (112 samples) along with protein data (RPPA) for 673 patients. More details on the abundance of data types' samples can be found in **Figure S11B**.

Data were downloaded from cBioPortal[1] (Gao et al., 2013). To explore all possibilities offered by the two datasets, we have used both of them to show different outcomes, METABRIC results are hereby showcased and TCGA results can be found in **Supplementary Material**.

2.3. Adapting Patient Profiles to a Logical Model

For this analysis, we gathered the following types of data: mutations, copy number variations, transcriptomics, proteomics and clinical data. Usually, mutations and copy number variations can be considered as discrete data and gene or protein expression data as continuous data. Two approaches for handling the data can be used in MaBoSS: (1) discrete data can be directly binarized, and (2) continuous data can either be binarized or normalized (expression values are modified so as to fit between 0 and 1). A logical model is personalized differently according to the type of data used. For instance, a deleterious mutation is integrated into the model by setting the corresponding node to 0 and ignoring the logical rule associated to it. For activating mutation, the node is set to 1. Another approach is to modify the transition rates (speed of activation or inactivation of a node, see section 2.1.2

[1] http://www.cbioportal.org/index.do

and **Figure S5**) according to the impact of the mutation or the level of gene or protein expression (further details in section 2.4).

In many mathematical models related to gene networks, some genes are often listed with a generic name and it is not always clear which gene is responsible of the reaction or if it rather refers to a family of genes (e.g., AKT for AKT1, AKT2, AKT3). Thus, before personalizing the models to patient data, a correspondence between model genes and data must be established and choices must be made on which genes to associate to the model's nodes. For our example, the complete table of correspondence of the model is available in our GitHub repository.

2.3.1. Processing of Discrete Data

Discrete data can be integrated in a straightforward manner through functional inference. From METABRIC database, we gathered mutations and copy number alterations.

2.3.1.1. Mutations

Based on the variant classification provided by the data, inactivating mutations (nonsense, frame-shift insertions or deletions and mutation in splice or translation start sites) are assumed to correspond to loss of function mutations and therefore the corresponding nodes of the model are forced to 0. Missense mutations are matched with OncoKB database (Chakravarty et al., 2017). For each mutation present in the database, an effect is assessed (gain or loss of function assigned to 1 and 0, respectively) with a corresponding confidence based on expert and literature knowledge. Mutations targeting oncogenes (resp. tumor-suppressor genes), as defined in the 2020+ driver gene prediction method (Tokheim et al., 2016), are assumed to be gain of function mutations (resp. loss of function) and therefore assigned to 1 (resp. 0). To rule potential passenger mutations out, each assignment requires a label of deleteriousness either from SIFT (Kumar et al., 2009) or from PolyPhen scores (Adzhubei et al., 2010).

2.3.1.2. Copy number alterations

For CNA integration, only amplifications (+2) and homozygous deletions (−2) (based on GISTIC processing Mermel et al., 2011) are considered, but this choice can be adapted to the focus of the study. Nodes corresponding to amplified genes are set to 1 and those associated with homozygous deletions are set to 0 in patient profiles. In our approach, we chose to discard CNA GISTIC variations with values −1 and +1 due to their low-confidence significance.

2.3.2. Processing of Expression Data

To be integrated into the logical model, continuous data must be either binarized or normalized between 0 and 1. To do so, gene expression data are first classified in three broad categories according to their distribution across samples: bimodal, unimodal, and zero-inflated distribution. Genes with different distributions are treated differently as summarized in **Figure 3**. Binarization and normalization methods different from the ones proposed here (e.g., Müssel et al., 2015; Jung et al., 2017) may also be used and directly integrated in the pipeline presented in the 2.4 section.

2.3.2.1. Distribution classification

Non-variant genes are discarded based on the admissibility test: the test verifies that the gene expression is included in a sufficient range of values compared to other genes (i.e., a gene's amplitude across the cohort above one tenth of median amplitude across all genes) and contains a sufficient number of non-zero values (i.e., at least 5% of non-zero values). In single-cell transcriptomics terminology, the latter corresponds to a low drop-out rate.

In order to classify the remaining genes, we identify bimodal patterns based on three distinct criteria: Hartigan's dip test of unimodality, Bimodality Index (BI) and kurtosis.

The dip test measures multi-modality in a sample using the maximum difference between empirical distribution and the best unimodal distribution, i.e., the one that minimizes this maximum difference (Hartigan and Hartigan, 1985). Values below 0.05 indicate a significant multi-modality. In PROFILE, this dip statistic is computed using the R package `diptest`.

The Bimodality Index (BI) evaluates the ability to fit two distinct Gaussian components with equal variance (Wang et al., 2009). Once the best 2-Gaussian fit is determined, along with the respective means μ_1 and μ_2 and common variance σ, the standardized distance δ between the two populations is given by

$$\delta = \frac{\mid \mu_1 - \mu_2 \mid}{\sigma} \tag{1}$$

and the BI is defined by

$$BI = [\pi(1-\pi)]^{\frac{1}{2}}\delta \tag{2}$$

where π is the proportion of observations in the first component. In PROFILE, BI is computed using the R package `mclust`.

Finally, the kurtosis method corresponds to a descriptor of the shape of the distribution, of its tailedness, or non-Gaussianity. A negative kurtosis distribution, especially, defines platykurtic (flattened) distributions, and potentially bimodal distributions. It has been proposed as a tool to identify small outliers subgroups or major subdivisions (Teschendorff et al., 2006). In our case, we focus on negative kurtosis distributions to rule out non-relevant bimodal distributions composed of a major mode and a very small outliers' group or a single outlier (an example of which can be seen in **Figure S7**).

Although dip test, BI and negative kurtosis criteria emerge as similar tools in the sense that they select genes whose values can be clustered in two distinct groups of comparable size, we choose to combine them in order to correct their respective limits and increase the robustness of our method (see bimodality test in **Figure 3C**). For that, we consider that all three conditions (Dip test, Bimodality Index and kurtosis) must be fulfilled in order for a gene to be considered as bimodal.

The thresholds of each test are inspired by those advocated in the papers presenting the tools individually. Dip test is a statistical test to which the classical 0.05 threshold has been chosen. In the article describing BI, authors explored a cut-off range between 1.1 and 1.5 and we chose 1.5 for the present work. Regarding kurtosis, the usual cut-off is 0, but since this criterion does not directly target bimodality, this criterion has

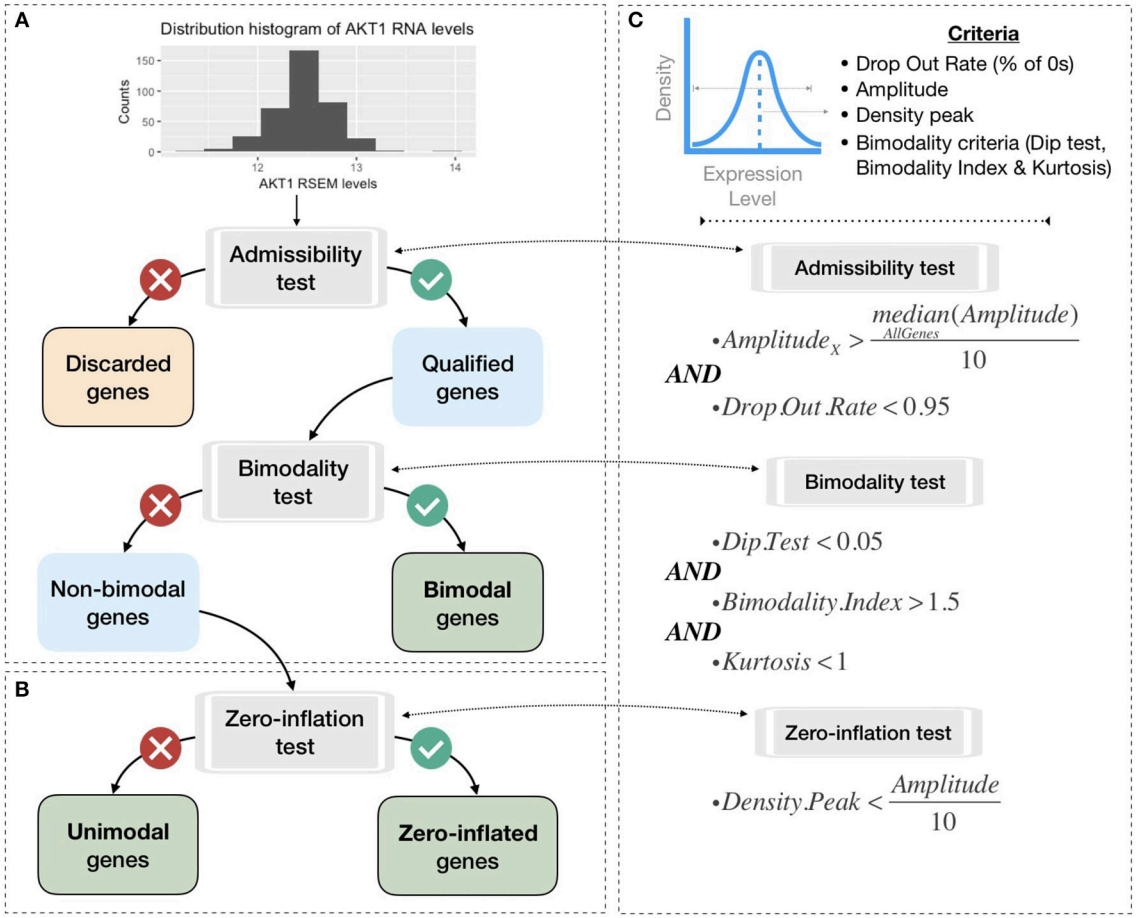


FIGURE 3 | Processing pipeline to classify genes in different categories based on their expression pattern across the cohort. **(A)** Tests to separate bimodal from non-bimodal genes before subsequent binarization. **(B)** Test to classify non-bimodal genes in unimodal or zero-inflated genes. **(C)** Statistical and logical content of the various tests used in **(A,B)**, thresholds have been taken from the papers presenting each tool and are more precisely justified in the Methods section.

been relaxed to $K < 1$. Several examples of the relative differences and complementarities between these criteria can be seen in **Figure S7**.

This method is enough to binarize continuous data as can be seen in **Figure S8**. However, to normalize continuous data, we need to further classify non-bimodal gene distributions among unimodal or zero-inflated, looking at the position of the distribution density peak. Then, based on this three-category classification of genes, we performed binarization and normalization processing as summarized in **Figure S8**.

Because the normalization of continuous data preserves more original information than its binarization, we will detail here only the normalization process. However, it should be noted that the preliminary classification of gene distributions into three distinct categories allows for a simple binarization (**Figure S8**).

Normalization functions are thus defined as follows:

$$\begin{aligned} Bin\colon OriginalValues &\to BinarizedValues \\ X &\mapsto Bin(X) \\ Norm\colon OriginalValues &\to NormalizedValues \\ X &\mapsto Norm(X) \end{aligned}$$

2.3.2.2. Bimodal genes processing: Gaussian mixture models

In PROFILE, a 2-component Gaussian mixture model is fitted using `mclust` R package resulting in a lower mode M_0 and a upper mode M_1 (**Figure 4**). Each data point X has a probability to belong to M_0 or M_1 such as

$$Prob(X_{gene_i,sample_j} \in M_{0,gene_i}) + Prob(X_{gene_i,sample_j} \in M_{1,gene_i}) = 1 \quad (3)$$

For these bimodal genes, the normalization processing is defined as:

$$Norm(X_{gene_i,sample_j}) = Prob(X_{gene_i,sample_j} \in M_{1,gene_i}) \quad (4)$$

2.3.2.3. Unimodal gene sigmoid normalization

For unimodal distributions, we transform data through a sigmoid function in order to maintain the most common pattern which is unimodal and nearly-symmetric. First of all, expression data are

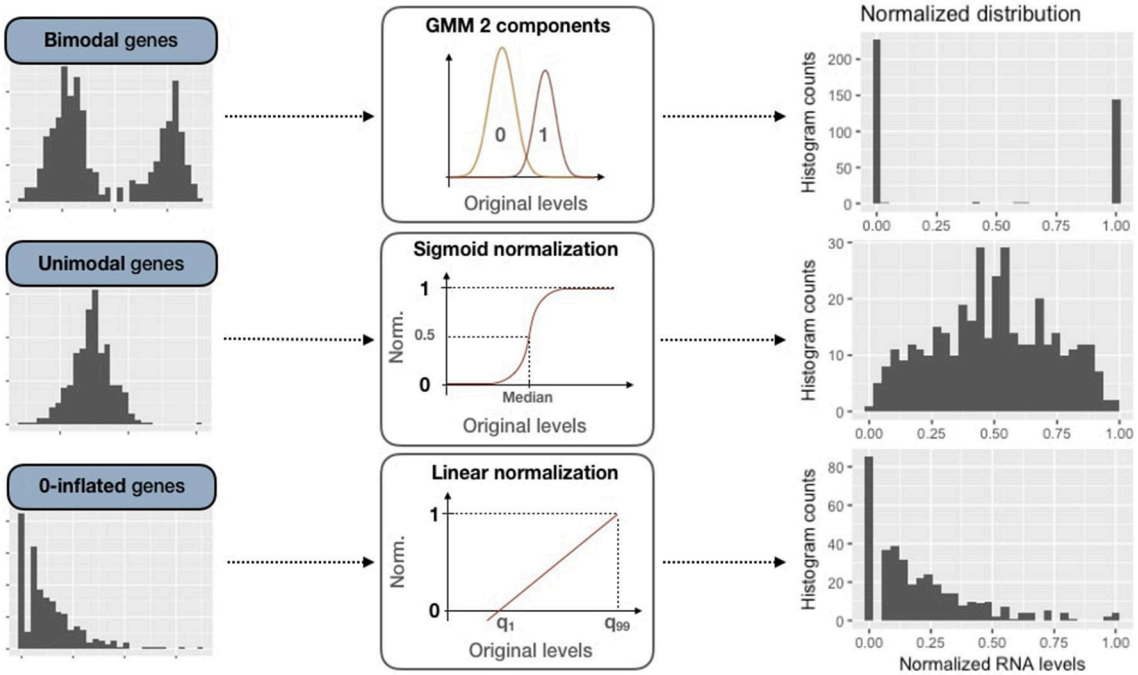


FIGURE 4 | Normalization methods for expression data of genes of all three categories (bimodal, unimodal and zero-inflated). First column panels show examples of original patterns from the three categories. Second column panels illustrate the processing methods used for normalization (GMM 2-component, sigmoid normalization and linear normalization). Third column panels correspond to normalized distributions.

centered around the median, which is more robust than using the mean regarding outliers:

$$X'_{gene_i,sample_j} = X_{gene_i,sample_j} - median_{gene_i}(X) \quad (5)$$

Then data are normalized through the sigmoid function:

$$Norm(X'_{gene_i,sample_j}) = \frac{1}{1 + e^{-\lambda X'_{gene_i,sample_j}}} \quad (6)$$

Since the slope of the function depends on λ, we adapt λ to the dispersion of initial data in order to maintain a significant dispersion in $[0, 1]$ interval: more dispersed unimodal distributions are mapped with a gentle slope, peaked distributions with a steep one. We map the median absolute deviation (MAD) on both sides of the median respectively to 0.25 and 0.75 to ensure a minimal dispersion of the mapping. First, the MAD is defined as:

$$MAD_{gene_i}(X) = median(\mid x_i - median_{gene_i}(X_{gene_i,sample_j}) \mid) \quad (7)$$

Therefore, to fulfill the proposed mapping, we solve:

$$\frac{1}{1 + e^{\pm \lambda MAD}} = \frac{1}{2} \mp \frac{1}{4}, \quad (8)$$

and derive:

$$\lambda = \frac{log_e(3)}{MAD} \quad (9)$$

Thus, we obtain data normalized in $[0, 1]$ for unimodal genes, as in **Figure 4**.

2.3.2.4. Zero-inflated genes sigmoid normalization

Zero-inflated genes are characterized by a distribution density peak (computed in PROFILE with the `density` function of `stats` R package) close to 0 (**Figure 3B**). For this case, we linearly transform the initial distribution in order to maintain the asymmetric original pattern:

$$Norm(X_{gene_i,sample_j}) = \frac{X_{gene_i,sample_j} - min_{gene_i}(X)}{max_{gene_i}(X) - min_{gene_i}(X)} \quad (10)$$

The transformation is applied to data between 1*st* and 99*th* quantiles to be more robust to outliers. Values below q_1 or above q_{99} are respectively assigned to 0 and 1.

2.3.2.5. Reference expression dataset

For the processing of expression data, two main options are available in PROFILE depending on what reference dataset is taken into account. We can either binarize/normalize genes based on distribution patterns across the whole cancer cohort or based on healthy patients. In the latter case, the type of gene distributions (bimodal, unimodal and zero-inflated) and the corresponding parameters (like inter-quartile range) are defined based on distribution patterns for healthy samples only, and the binarization/normalization is then applied on cancer patients. In the datasets under consideration in the present work, only the

TCGA RNA dataset includes healthy samples. Except otherwise stated, genes are processed based on cancer cohort and not based on healthy samples.

2.4. Personalization of Logical Models Using Patient Data

Personalization has been defined here as the specification of a logical model with data from a given patient: each patient has a personalized model tailored to his/her data, so that all personalized models are different specifications of the same logical model, using data from different patients (**Figure 1**). Based on MaBoSS formalism and the processed patient data, there are several possibilities to personalize a generic logical model with patient data as represented in **Figure 5**.

2.4.1. Activity of Model Nodes

One possibility to have patient-specific models is to force the value of the variables corresponding to the altered genes, i.e., constraining some model nodes to an inactive (0) or active (1) state. In order to constrain a node to 0 (resp. 1), the initial value of the node is set to 0 (resp. 1) and k_{up} (resp. k_{down}) to 0 to force the node to maintain its defined state. For instance, the effect of a p53 inactivating mutation can be modeled by setting the node TP53 in the model and its initial condition to 0 and ignoring the logical rule of TP53 variable. These modifications are referred to as node activity in the logical model. This constraint affects the simulation trajectories and consequently may shift the trajectories in the solution state space (referred to as the state transition graph, STG) leading to a change in probabilities of the resulting stable states (very often, these nodes are the ones representing biological phenotypes that are used as read-outs of the model) (Grieco et al., 2013; Remy et al., 2015).

2.4.2. Initial Conditions

Another possible strategy is to modify the initial conditions of the variables of the altered genes according to the results of the binarization/normalization. These initial conditions can capture different environmental and genetic conditions. Nevertheless, in the course of the simulation, these variables will be prone to be updated depending on their logical rules. These initial conditions can either be binary or continuous between 0 and 1, so both binarized and normalized profiles can be used. In the present study, we have only considered patients' expression data to be included as initial conditions, but PROFILE allows for more data types to be used as initial conditions.

2.4.3. Transition Rates

Finally, as MaBoSS uses Gillespie algorithm to explore the STG, data can be mapped to the transition rates of this algorithm. In the simplest case, all transition rates of the model are set to 1, meaning that all possible transitions are equally probable. Alternatively, it is possible to separate the speed of processes by setting the transition rates to different values to account for what is known about the reactions: more probable reactions will have a larger transition rate than less probable reactions (Stoll et al., 2012). For this, different orders of magnitude for these values can be used. They are set according to the activation status of the node (derived from normalized or binarized values) and an "amplification factor," designed to generate a higher relative difference in the transition rates, as follows:

$$k^{up}_{gene_i,sample_j} = AmplificationFactor^{2(Norm(X_{gene_i,sample_j})-0.5)} \quad (11)$$

$$k^{down}_{gene_i,sample_j} = \frac{1}{k^{up}_{gene_i,sample_j}} \quad (12)$$

Thus, if a gene has a value of 1 based on its RNA profile, its transition rate from 0 to 1 (resp. from 1 to 0) will be 10^2 (resp. 10^{-2}) with an amplification factor of 100.

Note that in the present study, we have only considered normalized patients' expression data to be included as transition rates (RNA for METABRIC data and RNA or Protein for TCGA data). The influence of the amplification factor on the results is discussed in **Section 1.6.2** and **Figure S10** (Supplementary Material). Based on this analysis, we chose an amplification factor of 100.

2.4.4. Synthetic Definition of Logical Model Personalization

We propose to summarize personalization methods in two different strategies (**Figure 5**). One one hand, applying *Strict Node Variants* (Strict NV) method, nodes for which data are available, are set to a given value for the whole simulation. For these nodes, logical rules are no longer in use, as they will always have a given value (0 or 1).

On the other hand, combining *Initial States* and *Transition Rates* modifications, we define a *Soft Node Variants* (Soft NV) method. Using this method, if a given node has a normalized value of 0.8 after data processing (based on proteins levels for instance), it will be initialized as 1 in 80% of the stochastic trajectories, its transition rate $k_{0\to1}$ will be increased (favoring its activation) and its transition rate $k_{1\to0}$ will be decreased (hampering its inactivation). These changes increase the probability that this node will remain in an activated state close to the one inferred from the patient's data, while maintaining the validity of its logical rule. Thus, Soft NV appears as a smoother way to shape logical models' simulations based on patient data.

2.4.5. Combinations of Data Types

The choice of which data types to include and where to map these data on the modeling framework is dependent on the goals of the study. If mutations, CNA and gene and protein expression data are provided for a given patient, one could include all these data types as follows: nodes corresponding to mutations and CNA could be used to specify model nodes (set to 0 or 1 if they are inhibiting or activating mutations or if they are homozygous deletions or amplifications), and transition rates could be modified to account for gene and protein expression levels.

Mapping different data types with different personalization methods avoids potential conflicts. However, combining different

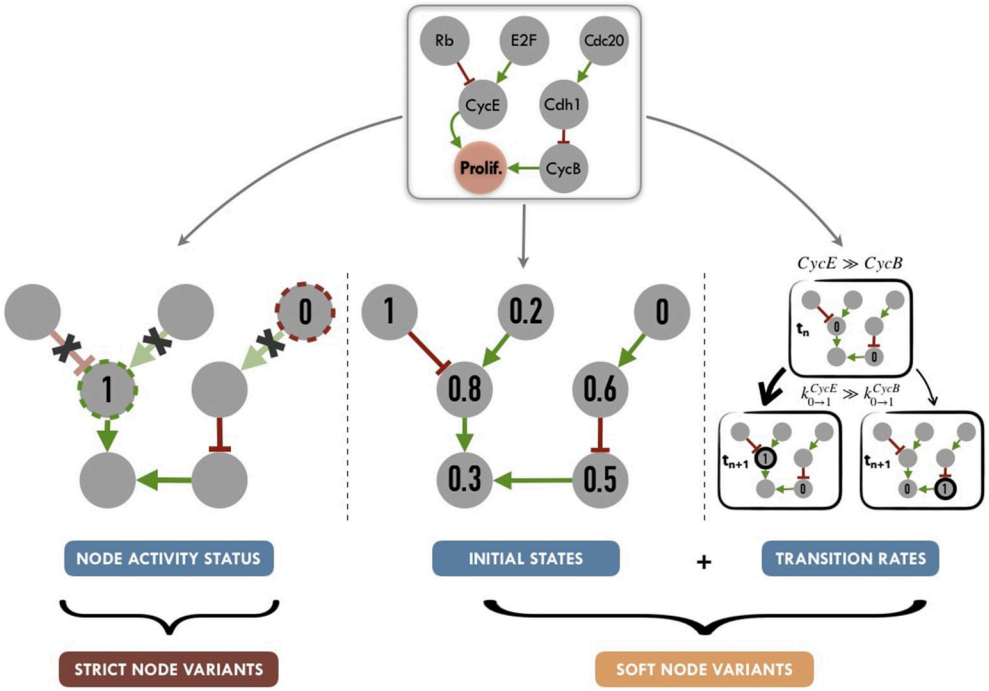


FIGURE 5 | Graphical summary of personalization methods for logical models: node activity status (nodes set to a given value for the full run of the simulation, without consideration for logical rules), initial states (nods initialized to a given proportion of 0 or 1 states) and transition rates (weighted stochastic transitions depending on the assumed activity level of nodes). Tuning node activity status, we define *Strict Node Variants* or Strict NV, i.e., forcing the activity (either present or absent) of a node. Tuning both initial states and transition rates, we define *Soft Node Variants* or Soft NV, *i.e*, nodes whose initial activation proportion is close to that of the patient and whose transition rates promote maintenance in its state of activity.

data types with the same personalization method raises some ambiguity issues. For instance, a gene can be inferred as a loss of function from the mutation data and can be found as amplified from CNA data. In this case, we consider that the information from mutations always overrides the information coming from CNA or binarized RNA/protein. Since both RNA and protein expression are available in the TCGA dataset, we explored the possibility to combine the two data types as follows: the RNA expression level is taken into account to define soft node variants only if there is no corresponding data in the protein dataset for that specific node. In the section 3.2, we present different choices that can be made according to the studied goals and data availability and in section 3.3, we analyze which combination is best suited to explain our patients' clinical data.

2.5. Comparison With Clinical Data

In order to assess the relevance of the different scenarios of model personalization, we investigate the correlation with biological and clinical factors.

For METABRIC dataset, signatures from the Molecular Signature Database (MSigDB) described in Liberzon et al. (2015) were used to classify the relevance of *Proliferation* and *Apoptosis* probabilities obtained from different personalization methods. We selected the Hallmarks "G2M Checkpoint" (resp. "Apoptosis"), a gene set composed of 200 genes (resp. 161) to correlate with the *Proliferation* (resp. *Apoptosis*) model probabilities. Genes used to personalize the models are excluded from the gene set, which reduces it to 185 (resp. 150) genes. Signature scores are then computed with the Gene Set Variation Analysis (GSVA) method, described in Hänzelmann et al. (2013) and implemented in `GSVA` R package. Correlations are assessed based on Spearman rank method and 95% confidence intervals are obtained by bootstrap ($n = 1,000$). For the METABRIC cohort, the patient's Nottingham prognostic index (NPI) and survival data are also gathered. NPI is a prognostic score based on clinical features such as tumor size, tumor grade and node status.

Regarding the survival data, there is data for all but one of the 1980 METABRIC patients. The overall survival time points are between 0 and 355 months with a median survival time of 283 months and 646 events (patients died of disease). Kaplan-Meier fits are obtained using the `survival` R package.

2.6. Availability

All the scripts and models are freely available on GitHub (https://github.com/sysbio-curie/PROFILE) and are distributed open source under the BSD 3-clause license. This repository can be referred to with its own DOI: (https://doi.org/10.5281/zenodo.1491229).

3. RESULTS

3.1. Breast Cancer Data Processing

Our framework has been applied to 2,509 breast cancer patients' molecular data that were collected from METABRIC. Patients'

data types include exome mutations, CNA and RNA expression as well as clinical data such as survival data. One thousand nine hundred and four patients of the 2,509 total have all these data types available (Curtis et al., 2012; Pereira et al., 2016) (**Figure S8A**). Data were processed as described in previous sections.

The logical model of cancer pathways (Fumiã and Martins, 2013) was chosen as a working example as it is a generic model with a relatively big number of nodes that span several pathways relevant to cancer. This model was initally used to study the effects of microenvironment conditions, to simulate the response to driver mutations in colorectal cancer progression and the effect of genes' perturbations as therapeutic targets (Fumiã and Martins, 2013).

Data from the METABRIC dataset that were relevant to the model were selected. Focusing on the 110 genes overlapping with nodes of the logical model, exome sequencing resulted in 2,659 mutations, of which 1,431 mutations were inferred as loss of function and 1,228 as gain of function. Besides, 634 mutations have unknown or silent effects and therefore were not considered. These 3,293 model-related mutations represent 19% of all mutations of the METABRIC dataset. Note that these numbers show the intersection of a generic model and a breast-cancer-specific dataset, so this percentage could be further increased by using a model with breast-specific pathways. Patients' profiles were found to have up to 7 mutations with most patients having only one assigned mutation. PIK3CA and TP53 were found to be the most frequently mutated genes.

For CNA data, patients' profiles had up to 19 perturbations, with a median number of 2. MYC gene was the most frequent gene with copy number alterations.

RNA expression data were processed and genes were separated in bimodal, unimodal and zero-inflated categories (**Figure 3** and section 2.3). All model-related genes in METABRIC cohort were found to be unimodal. Note that bimodal genes occur in several biologically meaningful situations like fusion genes such as ERG in prostate cancer or hormone genes such as ESR1 in breast cancer. We chose to explore the results of the METABRIC data with a model built specifically for breast cancer analysis (Zañudo et al., 2017) in order to assess the importance of including cancer-specific genes. Indeed, ESR1 is present in the breast-specific logical model analyzed in **Supplementary Material**.

The methods of binarization and normalization are applied to each data type according to the previously presented rules (**Figure 4** and **Figure S8**).

We further compared our binarization method to an existing tool, RefBool framework (Jung et al., 2017), using the same METABRIC dataset. This tool uses a set of reference distributions and it results in *p*-values for each sample and gene, assessing the significance of its putative binarization. Using 0.05 as a binarization threshold for RefBool *p*-values on the whole METABRIC RNA dataset (1904 samples and 24368 genes), around 4.4 million values were binarized (9.5% of the total). All of these binarizations resulted in active nodes and thus set to 1. Notably, RefBool was designed to use a reference dataset to binarize new data. Due to the lack of a reference healthy dataset in METABRIC, the whole dataset has been used as its own reference: each gene was compared to the distribution of that gene across all samples. Comparatively, our method results in 2.8 million of binarized values (6.1% of the total), respectively 4.2% of 1 s and 1.9% of 0 s. There seems to be a trend for RefBool in METABRIC dataset to emphasize positive outliers at the expense of negative ones, even for roughly symmetric unimodal distributions (**Figure S18**). Some examples of this dataset can be studied in **Supplementary Material**, together with the analysis on TCGA dataset, that bears healthy samples, and should be a better showcase to RefBool capabilities (**Figure S19**).

3.2. Personalization of a Generic Logical Cancer Model With Breast Cancer Data

We proceeded to personalize the logical model using different types of data and several data integration methods, such as on the activity of the nodes, the initial conditions and the transition rates. The effect of integrating different data at different levels of the model are represented by different phenotypes' distributions that can be used to study the respective effects of model personalization methods in **Figure 6**. Note that the probabilities for the wild type conditions are 0.019 for *Proliferation* and 0.906 for *Apoptosis* and are represented as a black dashed vertical line in **Figure 6**.

Using mutation data as a forced activity (either present or absent) of a node of the model (termed *Strict Node Variants* or Strict NV throughout the text), resulted in the distribution of *Proliferation* probabilities around the value 0.05 and the distribution of *Apoptosis* probabilities around two values (0.5 and 0.85) in **Figure 6** (upper panels, case 1). It is important to note that as these data are discrete and sparse, this causes the *Proliferation* distribution to be quite sharp. The distribution becomes smoother when exome mutations and CNA are both considered as Strict NV of the model and peaked around two values (0.05 and 1 for *Proliferation* and 0.5 and 0.85 for *Apoptosis*), as shown on **Figure 6**, case 2. Using CNA information as *Soft Node Variants* (Soft NV) and mutation as Strict NV, the highly proliferative mode is slightly decreased, consistent with less stringent constraints (**Figure 6**, case 3). When only RNA expression levels are used as modified transition rates, the resulting distribution of phenotypes' probabilities is more dispersed (**Figure 6**, case 4) and only one lowly proliferative peak appears. Adding mutations information as Strict NV does not shift the probabilities' distributions (case 5). Lastly, when we consider mutations and CNA as nodes' activity and RNA expression levels as modified transition rates, it results in a combination of the previously observed patterns (**Figure 6**, case 6).

Nevertheless, the generic logical model we use here does not take into account key genes in breast cancer progression such as hormone receptors and their associated signaling networks. As previously mentioned, a breast-cancer-specific model (Zañudo et al., 2017) was investigated using the same METABRIC dataset to personalize breast patient-specific models with similar trends to those of the generic model's study (**Figure S12**). Zañudo et al. (2017) model generates narrower distributions and therefore less discriminating probabilities from one patient to another, which is mainly due to the fact that it captures less information due to its lower number of nodes (especially with sparse data such as mutations). For these reasons, and having in mind the

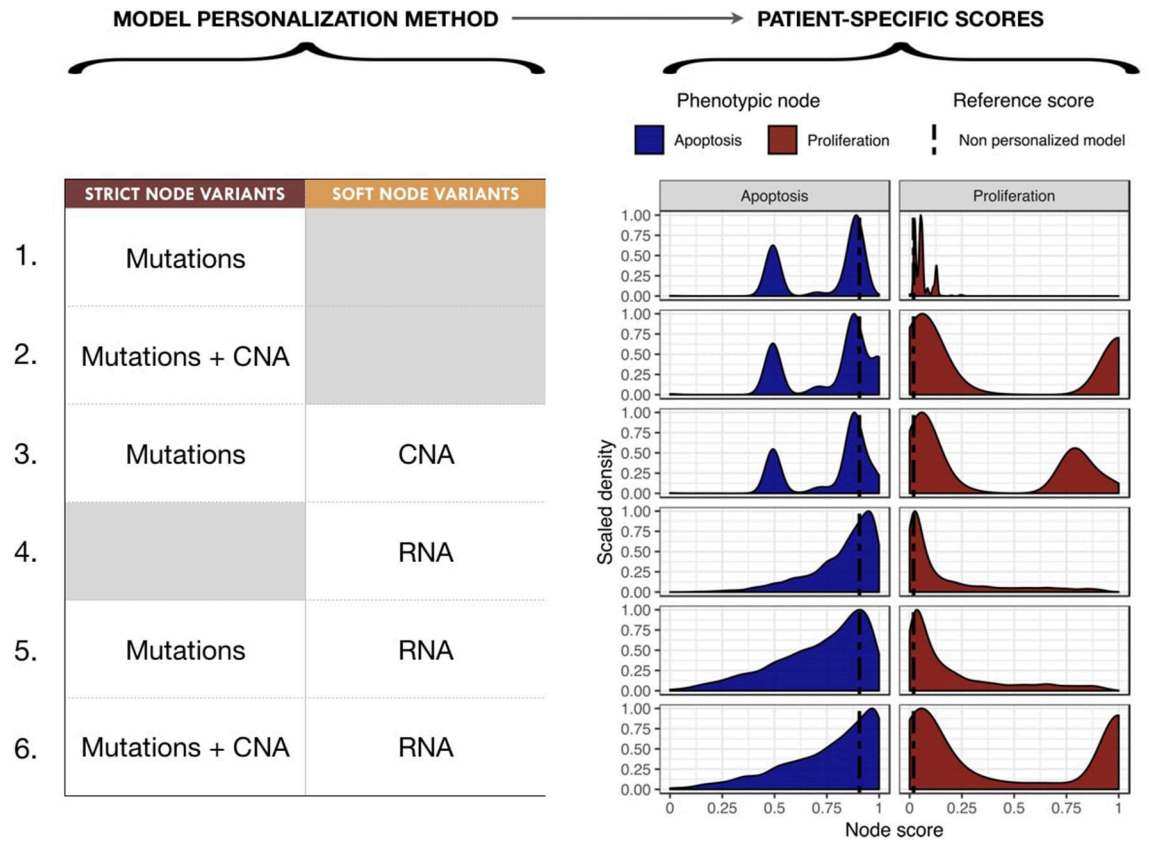


FIGURE 6 | Impact of different model personalization methods on the distribution of phenotypic nodes *Proliferation* and *Apoptosis*, using a generic cancer model and METABRIC data. On the left, description of data types used as *Strict* or *Soft Node Variants* to personalize the model resulting in different phenotypic probabilities' distributions across the cohort, as shown on the right. Dashed lines correspond to the probabilities of the phenotypes obtained from the original model without any personalization: 0.019 for *Proliferation* and 0.906 for *Apoptosis*.

methodological scope of present work, we will focus on the discussion on results of the more comprehensive generic model.

In order to present the use of one model with more than one dataset, PROFILE method was also done and analyzed using TCGA molecular data on Fumiã and Martins (2013) generic model (**Figure S14**).

Figures such as **Figure 6** are useful to identify the integration of which data in which part of the model has a greater impact in the change in phenotypes' distributions, but say little about the biological relevance of these distributions. To further investigate which combinations of methodology provides better biological or clinical insights, we compared these models' results to several signatures or clinical factors used in breast cancer studies.

3.3. Selecting Personalization Methods Using Correlation of Phenotypic Probabilities to Signature Scores

To classify the relevance of the six personalization methods presented in the previous section, we studied the correlations of the probabilities of the model phenotypes with representative signatures of the same phenotypic processes. This methodology allows to classify the different personalization methods and to study which one is better suited to describe the diversity of patients when tailoring a given model to a given dataset.

The Spearman rank correlations of the *Apoptosis* probabilities from personalized models with the RNA-based "Apoptosis" signature defined in the Hallmarks (Liberzon et al., 2015) gene set was computed (**Figure 7A**). Sparse binary data (when using mutations or CNA data) appear to be a poor choice to recover a consistent Apoptosis probability with the logical models (cases 1, 2 and 3). Only models personalized with RNA data as Soft NV are able to mimic an Apoptosis behavior consistent with the signature.

When comparing the *Proliferation* probabilities from the models to the Hallmarks' "G2M Checkpoint" signature (**Figure 7A**), personalized models are able to capture consistent behavior regardless of the type of data used as input. Nevertheless, the best Spearman rank correlations coefficients used as classifiers singled out the cases that use RNA as Soft NV (cases 4, 5 and 6), specially when the activity of nodes was fixed by mutations and transition rates by RNA values (case 5, mean Spearman's ρ of 0.61). In spite of their smaller correlation, the first three cases are also of interest since they only make use of originally sparse and discrete information: mutations and CNA data used as Strict and/or Soft NV. For instance, in case 3, using

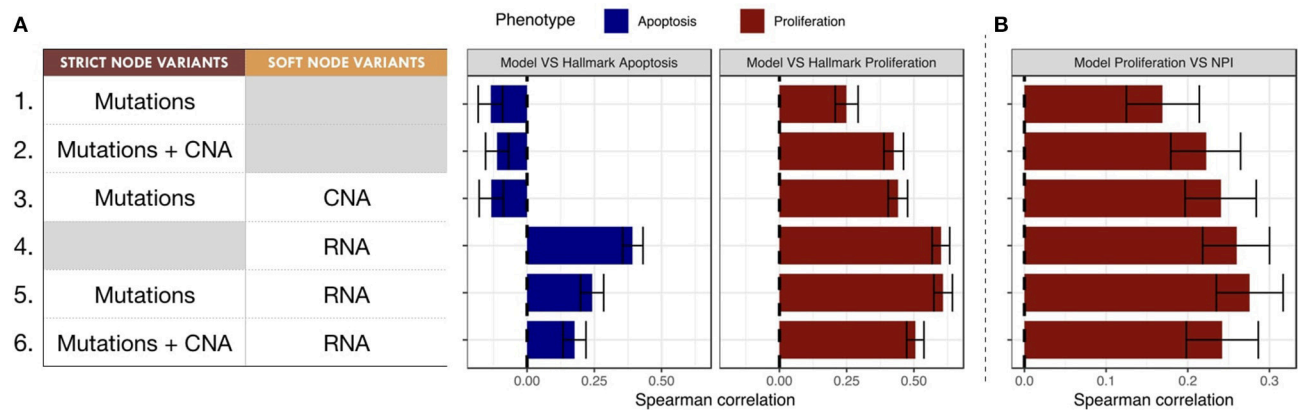


FIGURE 7 | Biological and clinical classification of phenotypic probabilities from personalized models. **(A)** Model personalization methods used and the corresponding Spearman rank correlation between phenotypic probabilities from personalized model and the corresponding Hallmark signatures score based on RNA gene sets. **(B)** Spearman rank correlation between personalized *Proliferation* probabilities and the Nottingham Prognostic Index (NPI) score based on clinical features (size and grade of tumor, node status).

mutations as Strict NV and CNA as Soft NV, personalized models are able to retrieve 44% of proliferation information contained in RNA-based "G2M Checkpoint" signature (**Figure 7A**, case 3).

Similarly, when comparing the probabilities of the *Proliferation* phenotype to NPI scores (**Figure 7B**), a purely clinical index that is not based on omics data, we observe the same trends for correlations, but with decreased coefficients. This supports the potential of these personalized models to partially identify clinical information as discussed in the survival data in section 3.5.

3.4. Clinical Subgrouping of Patients' Specific Model Outputs

Next, we studied the relationship of our patients' specific model probabilities to the PAM50 subgrouping, defined by the expression of 50 genes (Parker et al., 2009). For this, *Proliferation* probabilities from the personalized models were compared across subtypes (**Figures 8A–C**).

Using only mutations and CNA (**Figure 8A**), two different patterns may be observed: Basal, Her2 and Luminal B patients have balanced *Proliferation* bimodal probabilities with both lowly and highly proliferative patients. The second pattern involves Claudin-low, Luminal A and Normal-like patients that are mainly lowly proliferative with a smaller highly-proliferative mode. This grouping of subtypes, based on distribution trends, is consistent with the distinct proliferative behaviors of breast-cancer subtypes as described in Prat and Perou (2011): although similar in some aspects Luminal subtypes are distinguished by the more proliferative aspect of Luminal B; Basal and Her2 subtypes are also considered as aggressive tumors in contrast to Luminal A and Normal-like; Claudin-low subtypes have mixed behaviors depending on conditions but are usually described as lowly proliferative *in vivo*. The trends captured by the model are therefore consistent with clinical knowledge.

When personalizing logical models with RNA but no CNA (**Figure 8B**), only the proliferative nature of the Basal subtype seems to be well described, even when using mutation data. When combining RNA and CNA data (**Figure 8C**), the previously described clinical trends are again observed with clearer distinctions between subtypes.

In order to provide a reference of subtyping using omics data, a Principal Component Analysis (PCA) was performed taking into account the RNA expression levels of the 114 genes related to all nodes of the model (**Figure 8D**). The first principal component (PC1) of this PCA captured the different molecular subtypes and sequentially separated different subtypes (Luminal A, Luminal B, Her2 and Basal). This analysis shows a smoother and more linear distribution of the different subtypes, while personalized models seem to assign them more discrete patterns.

3.5. Survival Analyses of Patients' Specific Model Outputs

As a follow-up to the correlation studies of phenotypes' probabilities and clinical NPI scores, METABRIC survival data were correlated to the *Proliferation* and *Apoptosis* probabilities. For the survival analysis, thresholds needed to be set for the probabilities for each phenotype in order to separate between two groups: high and low. These thresholds were defined using the median for each phenotype probability across the cohort. Thus, each patient was grouped into two groups (high or low) for each phenotype (*Proliferation* or *Apoptosis*).

Studying simulation results from case 3 (mutations as Strict and CNA as Soft NV), thresholds of 0.12 and 0.87 were determined for *Proliferation* and *Apoptosis* phenotypes respectively. Kaplan—Meier plot (Kaplan and Meier, 1958) for *Proliferation* low and high probabilities' groups were significantly different (log-rank test, $p = 2.05e^{-11}$) and low proliferative patients' models had better prognostic than the high ones (**Figure 9A**). When considered as a continuous biomarker, *Proliferation* appeared significant in a Cox model with a p-value of $p = 2.13e^{-8}$.

Similarly, Kaplan—Meier plot for *Apoptosis* low and high probabilities' groups were significantly different (log-rank test, $p = 8.82e^{-8}$) and high apoptotic patients' models had better

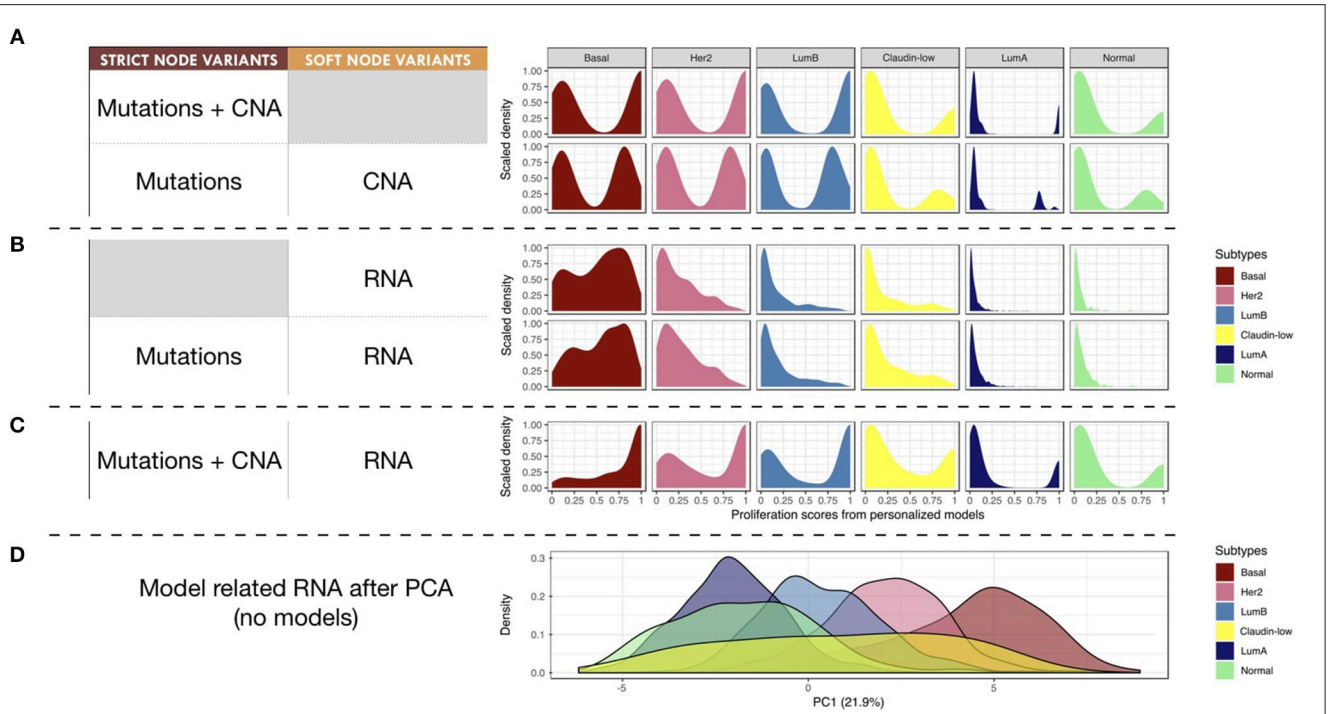


FIGURE 8 | Breast-cancer subtypes and personalized logical models. **(A–C)** Patterns of *Proliferation* probabilities per subtype with different personalization methods results in different grouping of subtypes. **(D)** Densities of breast-cancer subtypes along first principal component, PCA transformation based on METABRIC RNA data limited to the 114 genes described in the logical model ; percentage of variance explained by PC1 in parenthesis.

prognostic than the low ones (**Figure 9B**). When considered as a continuous biomarker, *Apoptosis* appeared significant in a Cox model with a p-value of $1.09e^{-8}$. The observation of survival curves for high apoptotic or low proliferative patients' models having a much better prognostic than the opposite phenotypes (**Figures 9A,B**) is in accordance with the underlying cancer biology and is an implicit validation on the relevance of the model and its simulations.

We next combined both thresholds to separate patients in four groups (high and low *Proliferation* and high and low *Apoptosis*) (**Figure 9C**) that was also significantly different (log-rank test, p-value of $9.57e^{-14}$). Using this combination, the best prognosis was for patients' models with low *Proliferation* and high *Apoptosis* and the worst prognosis was associated to patients' models with high *Proliferation* and low *Apoptosis*. Groups with the other labels (either high *Proliferation* and high *Apoptosis* or low *Proliferation* and low *Apoptosis*) had mild prognoses. This observed behavior is fully consistent with the expected influence of proliferation and apoptosis in cancer prognosis. Thus, using sparse and binary data, we show that personalized logical models result in a meaningful stratification of patients.

Next, based on **Figure 7**, the most effective personalization method was selected (case 5 using mutations as Strict and RNA as Soft NV) and its survival analysis had similarly consistent behaviors (**Figure 10**). Nevertheless, using only RNA as Soft NV (case 4 of **Figure 7**), *Proliferation* remains very significantly correlated with survival data but *Apoptosis* is not (**Figure S16**), supporting the importance of mutations data to retrieve biologically consistent behaviors.

Based on **Figures 7–10** we conclude that for an optimal integration of the data available in this logical model, the best combinations are to binarize mutations and treat them as Strict NV, and to integrate RNA as Soft NV. Replacing RNA with CNA data results also in largely consistent behaviors with sparser data.

We conclude that our personalization protocol is useful to build data-tailored models that can capture patient-specific phenotypes' behaviors which correlate to survival data.

4. DISCUSSION

In order to reach its full potential, personalized medicine needs precise mathematical models, and this will only be achieved with models tailored to the data for a given patient. These patient-specific models can be of great help to study patient-tailored drug combinations or the different drug responses in a group of patients with similar profiles and to advice the clinical oncologist as to the optimal treatment to choose for a given patient. The methodology presented here is a first step toward the personalization of a logical model to different patient profiles such that their results can be matched to clinical data and patients' subgrouping.

Our PROFILE framework is able to use different data types (mutation, CNA and gene and protein expression data) and incorporate them at different levels of the logical modeling formalism. The personalization strategies presented here have been compared to well-established signatures and NPI score, and the outcomes of these patient-specific models have shown

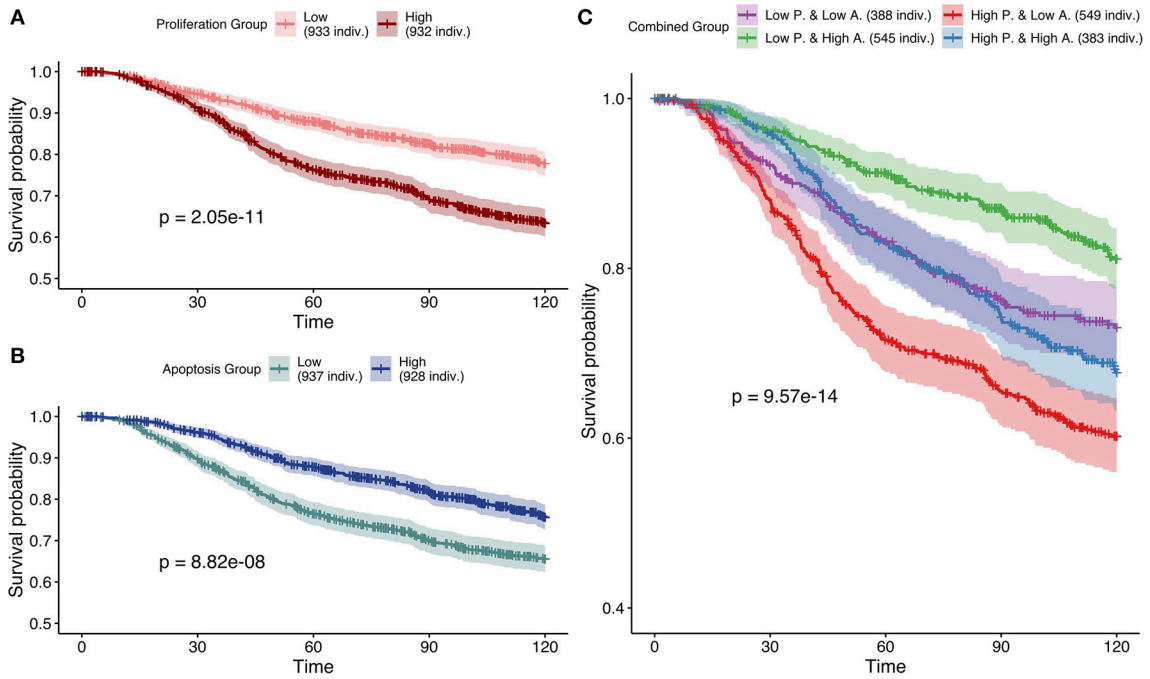


FIGURE 9 | Survival analyses of METABRIC samples from which exome mutations are used as *Strict Node Variants* and CNA as *Soft Node Variants* in the model (case 3). All *p*-values are derived from a log-rank test. **(A)** Survival curves with high and low *Proliferation* groups. **(B)** Survival curves with high and low *Apoptosis* groups. **(C)** Survival curves with combined groups.

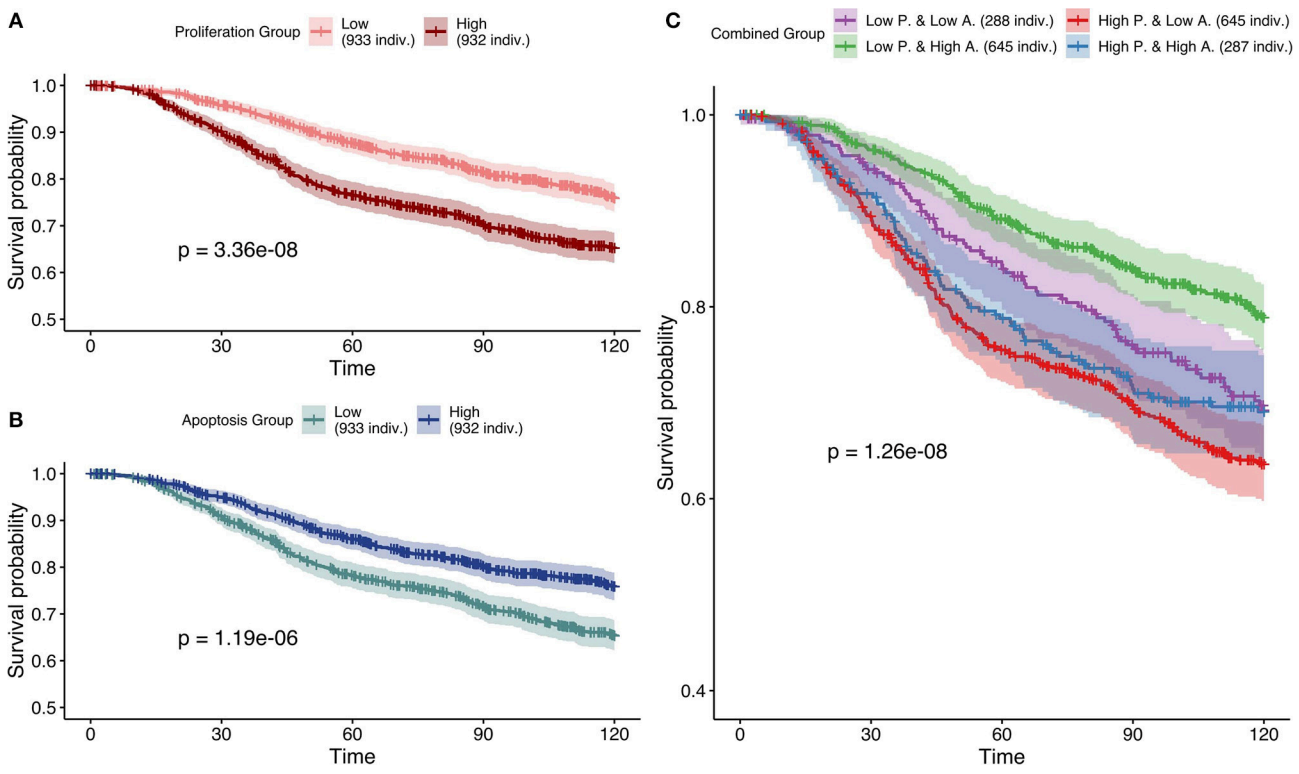


FIGURE 10 | Survival analyses of METABRIC samples from which exome mutations are used as *Strict Node Variants* and RNA as *Soft Node Variants* in the model (case 5). All *p*-values are derived from a log-rank test. **(A)** Survival curves with high and low *Proliferation* groups. **(B)** Survival curves with high and low *Apoptosis* groups. **(C)** Survival curves with combined groups.

to correlate well with clinical data. Any other relevant clinical measure could be used, especially more specific features corresponding to molecular mechanisms studied in the models. Notably, some choices on which data to include in the specification of the models are better than others when studying the correlation of the phenotypic probabilities of the logical model to signatures or the model ability to differentiate patients by prognostic outcome. To summarize, associating genetic mutations with the most stringent personalization method (i.e., Strict NV, constraining activity of nodes to either 0 or 1) and variation of copy number and expression levels with more permissive and stochastic personalization methods (i.e., Soft NV, intervening in initial states and transition rates) can be seen as biologically consistent. It is indeed expected that a genetic mutation can have a very strong and lasting effect that makes the gene independent of any regulation such as in the loss of function mutations. Conversely, the RNA expression level will affect the activity level of the genes but may not alter its regulation.

Using our PROFILE methodology, we are able to provide guidelines regarding the patient-data personalizations of logical models. Firstly, it is important to consider the nature of the node (gene or protein) in order to match the proper data type to the node. In the generic model used in our study, most of the nodes are supposed to be proteins, therefore it would be advisable to focus on protein data, which is unfortunately unavailable in the METABRIC dataset. In any case, the proposed framework could be easily adapted to the ideal case where each node would have a well-defined nature and a proper mapping of the corresponding data types. It is important to note that in the context of phospho-proteomic data (like RPPA's phosphosites), highly phosphorylated species can correspond to an inactive state that must be taken into consideration as mentioned in **Supplementary Materials** with TCGA data.

Secondly, healthy samples should be used if they are available in the dataset. Using an independent healthy samples for RNA normalization in TCGA dataset not only improved the correlation performances (**Figure S15**, case 4) but also the qualitative trend of the results (**Figure S17**). It can be seen that using healthy samples instead of cancer samples as a reference for RNA normalization results in a significant shift of the distribution toward high *Proliferation* model probabilities (**Figure S17**).

Thirdly, to improve the results of personalized logical models, the model used must be big enough, but also cover specificities of the cancer under study. Models should not be too generic, as they should include important read-outs of cancer types such as AR for prostate or ER and BRCA1 for breast cancer allowing them to better separate cancer subgroups. Also, they should include a sufficiently meaningful number of genes in order to be able to differentiate among patients.

In order to achieve clinically relevant models, it will be necessary to bring together the best of both worlds: large models able to integrate most alterations of common cancer pathways (e.g., DNA repair) and cancer-specific nodes (e.g., hormone receptors) able to explain the particular behavior of each cancer.

As perspectives, we plan to explore methods that will allow to use the solutions of the logical model for patient-specific studies. One possibility that would allow for personalized drug treatments is to integrate drug interactions in these personalized models, uncovering patient-specific drug targets whose behaviors might depend on environmental conditions. Another possibility that would enable a better patient stratification is to compute the Hamming distance of a binarized profile of a patient with each of the stable states obtained by the non-personalized model. That way, a patient can be considered "closer" to a given phenotype, such as *Proliferation*, *Apoptosis* or *Senescence*, etc. This approach raises problems such as how to treat attractors such as limit cycles, which are usually found in logical models, since this comparison can only be done on stable state solutions. We have started exploring this possibility (Cohen et al., 2015) and some work has been done by other groups in this direction (Dorier et al., 2016).

In conclusion, our PROFILE methodology allows to build precise mathematical models that captures the heterogeneity of patients profiles and their diverse behaviors. These logical models, which are properly specified with patient information, would enable clinicians to test personalized drugs combinations or therapeutic strategies *in silico* and pave the way to precision medicine.

AUTHOR CONTRIBUTIONS

LC and EB designed the project. JB, LC, PT, and AM participated in the conceptualization of the methodology. PT and JB designed the methods for the generation of patients' profiles from datasets. JB set up the pipeline for binarization and normalization of the data, generated patients' profiles from datasets, adapted these to the model and performed the personalization of models. PT and AM selected and analyzed the logical models. Manuscript was written by JB, AM, and LC and all authors read and edited it.

FUNDING

This work received funding from the European Union Horizon 2020 research and innovation program under grant agreement No. 668858 (PrECISE project). JB is supported by an AMX scholarship from the French Ministry of Superior Education and Research.

ACKNOWLEDGMENTS

We would like to thank Aurélien Latouche for critical reading of the manuscript and for fruitful discussions on the survival analysis.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphys.2018.01965/full#supplementary-material

REFERENCES

Abou-Jaoudé, W., Chaves, M., and Gouzé, J.-L. (2011). A theoretical exploration of birhythmicity in the p53-mdm2 network. *PLoS ONE* 6:e17075. doi: 10.1371/journal.pone.0017075

Abou-Jaoudé, W., Traynard, P., Monteiro, P. T., Saez-Rodriguez, J., Helikar, T., Thieffry, D., et al. (2016). Logical modeling and dynamical analysis of cellular networks. *Front. Genet.* 7:94. doi: 10.3389/fgene.2016.00094

Adzhubei, I. A., Schmidt, S., Peshkin, L., Ramensky, V. E., Gerasimova, A., Bork, P., et al. (2010). A method and server for predicting damaging missense mutations. *Nat. Methods* 7, 248–249. doi: 10.1038/nmeth0410-248

Arshad, O. A., and Datta, A. (2017). Towards targeted combinatorial therapy design for the treatment of castration-resistant prostate cancer. *BMC Bioinformatics* 18:134. doi: 10.1186/s12859-017-1522-2

Calzone, L., Tournier, L., Fourquet, S., Thieffry, D., Zhivotovsky, B., Barillot, E., et al. (2010). Mathematical modelling of cell-fate decision in response to death receptor engagement. *PLoS Comput. Biol.* 6:e1000702. doi: 10.1371/journal.pcbi.1000702

Cancer Genome Atlas Network (2012). Comprehensive molecular portraits of human breast tumours. *Nature* 490, 61–70. doi: 10.1038/nature11412

Chakravarty, D., Gao, J., Phillips, S., Kundra, R., Zhang, H., Wang, J., et al. (2017). OncoKB: a precision oncology knowledge base. *JCO Precis. Oncol.* 1, 1–16. doi: 10.1200/PO.17.00011

Chaouiya, C., Bérenguier, D., Keating, S. M., Naldi, A., van Iersel, M. P., Rodriguez, N., et al. (2013). SBML qualitative models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools. *BMC Syst. Biol.* 7:135. doi: 10.1186/1752-0509-7-135

Chaouiya, C., Naldi, A., and Thieffry, D. (2012). "Logical modelling of gene regulatory networks with GINsim," in *Bacterial Molecular Networks. Methods in Molecular Biology (Methods and Protocols)*, Vol. 804, eds J. van Helden, A. Toussaint, and D. Thieffry (New York, NY: Springer).

Cho, S.-H., Park, S.-M., Lee, H.-S., Lee, H.-Y., and Cho, K.-H. (2016). Attractor landscape analysis of colorectal tumorigenesis and its reversion. *BMC Syst. Biol.* 10:96. doi: 10.1186/s12918-016-0341-9

Ciriello, G., Gatza, M. L., Beck, A. H., Wilkerson, M. D., Rhie, S. K., Pastore, A., et al. (2015). Comprehensive molecular portraits of invasive lobular breast cancer. *Cell* 163, 506–519. doi: 10.1016/j.cell.2015.09.033

Cohen, D. P. A., Martignetti, L., Robine, S., Barillot, E., Zinovyev, A., and Calzone, L. (2015). Mathematical modelling of molecular pathways enabling tumour cell invasion and migration. *PLoS Comput. Biol.* 11:e1004571. doi: 10.1371/journal.pcbi.1004571

Curtis, C., Shah, S. P., Chin, S.-F., Turashvili, G., Rueda, O. M., Dunning, M. J., et al. (2012). The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature* 486, 346–352. doi: 10.1038/nature10983

Dorier, J., Crespo, I., Niknejad, A., Liechti, R., Ebeling, M., and Xenarios, I. (2016). Boolean regulatory network reconstruction using literature based knowledge with a genetic algorithm optimization method. *BMC Bioinformatics* 17:410. doi: 10.1186/s12859-016-1287-z

Faure, A., Naldi, A., Chaouiya, C., and Thieffry, D. (2006). Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle. *Bioinformatics* 22, e124–e131. doi: 10.1093/bioinformatics/btl210

Fey, D., Halasz, M., Dreidax, D., Kennedy, S. P., Hastings, J. F., Rauch, N., et al. (2015). Signaling pathway models as biomarkers: patient-specific simulations of JNK activity predict the survival of neuroblastoma patients. *Sci. Signal.* 8, ra130–ra130. doi: 10.1126/scisignal.aab0990

Fumiã, H. F., and Martins, M. L. (2013). Boolean network model for cancer pathways: predicting carcinogenesis and targeted therapy outcomes. *PLoS ONE* 8:e69008. doi: 10.1371/journal.pone.0069008

Gao, J., Aksoy, B. A., Dogrusoz, U., Dresdner, G., Gross, B., Sumer, S. O., et al. (2013). Integrative analysis of complex cancer genomics and clinical profiles using the cbioportal. *Scie. Signal.* 6, pl1. doi: 10.1126/scisignal.2004088

Gérard, C., and Goldbeter, A. (2016). Dynamics of the mammalian cell cycle in physiological and pathological conditions. *Wiley Interdiscip. Rev. Syst. Biol. Med.* 8, 140–156. doi: 10.1002/wsbm.1325

Grieco, L., Calzone, L., Bernard-Pierrot, I., Radvanyi, F., Kahn-Perlès, B., and Thieffry, D. (2013). Integrative modelling of the influence of MAPK network on cancer cell fate decision. *PLoS Comput. Biol.* 9:e1003286. doi: 10.1371/journal.pcbi.1003286

Hänzelmann, S., Castelo, R., and Guinney, J. (2013). Gsva: gene set variation analysis for microarray and rna-seq data. *BMC Bioinformatics* 14, 7. doi: 10.1186/1471-2105-14-7

Hartigan, J. A., and Hartigan, P. M. (1985). The dip test of unimodality. *Ann. Stat.* 13, 70–84. doi: 10.1214/aos/1176346577

Helikar, T., Konvalina, J., Heidel, J., and Rogers, J. A. (2008). Emergent decision-making in biological signal transduction networks. *Proc. Natl. Acad. Sci. U.S.A.* 105, 1913–1918. doi: 10.1073/pnas.0705088105

Hidalgo, M. R., Cubuk, C., Amadoz, A., Salavert, F., Carbonell-Caballero, J., and Dopazo, J. (2017). High throughput estimation of functional cell activities reveals disease mechanisms and predicts relevant clinical outcomes. *Oncotarget* 8, 5160. doi: 10.18632/oncotarget.14107

Hofree, M., Shen, J. P., Carter, H., Gross, A., and Ideker, T. (2013). Network-based stratification of tumor mutations. *Nat. Methods* 10, 1108. doi: 10.1038/nmeth.2651

Huang, S., Ernberg, I., and Kauffman, S. (2009). Cancer attractors: a systems view of tumors from a gene network dynamics and developmental perspective. *Semin. Cell Dev. Biol.* 20, 869–876. doi: 10.1016/j.semcdb.2009.07.003

Jung, S., Hartmann, A., and Del Sol, A. (2017). RefBool: a reference-based algorithm for discretizing gene expression data. *Bioinformatics* 33, 1953–1962. doi: 10.1093/bioinformatics/btx111

Kaplan, E. L., and Meier, P. (1958). Nonparametric estimation from incomplete observations. *J. Am. Stat. Assoc.* 53, 457–481. doi: 10.1080/01621459.1958.10501452

Kumar, P., Henikoff, S., and Ng, P. C. (2009). Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm. *Nat. Protoc.* 4, 1073. doi: 10.1038/nprot.2009.86

Le Novère, N. (2015). Quantitative and logic modelling of molecular and gene networks. *Nat. Rev. Genet.* 16, 146–158. doi: 10.1038/nrg3885

Liberzon, A., Birger, C., Thorvaldsdóttir, H., Ghandi, M., Mesirov, J. P., and Tamayo, P. (2015). The molecular signatures database hallmark gene set collection. *Cell Syst.* 1, 417–425. doi: 10.1016/j.cels.2015.12.004

Mermel, C. H., Schumacher, S. E., Hill, B., Meyerson, M. L., Beroukhim, R., and Getz, G. (2011). GISTIC2.0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers. *Genome Biol.* 12:R41. doi: 10.1186/gb-2011-12-4-r41

Montagud, A., Traynard, P., Martignetti, L., Bonnet, E., Barillot, E., Zinovyev, A., et al. (2017). Conceptual and computational framework for logical modelling of biological networks deregulated in diseases. *Brief. Bioinformatics*. doi: 10.1093/bib/bbx163. [Epub ahead of print].

Müssel, C., Schmid, F., Blätte, T. J., Hopfensitz, M., Lausser, L., and Kestler, H. A. (2015). Bitrina—multiscale binarization and trinarization with quality analysis. *Bioinformatics* 32, 465–468. doi: 10.1093/bioinformatics/btv591

Novák, B., and Tyson, J. J. (2004). A model for restriction point control of the mammalian cell cycle. *J. Theor. Biol.* 230, 563–579. doi: 10.1016/j.jtbi.2004.04.039

Parker, J. S., Mullins, M., Cheang, M. C., Leung, S., Voduc, D., Vickery, T., et al. (2009). Supervised risk predictor of breast cancer based on intrinsic subtypes. *J. Clin. Oncol.* 27, 1160–1167. doi: 10.1200/JCO.2008.18.1370

Pereira, B., Chin, S.-F., Rueda, O. M., Vollan, H.-K. M., Provenzano, E., Bardwell, H. A., et al. (2016). The somatic mutation profiles of 2,433 breast cancers refines their genomic and transcriptomic landscapes. *Nat. Commun.* 7:11479. doi: 10.1038/ncomms11479

Prat, A., and Perou, C. M. (2011). Deconstructing the molecular portraits of breast cancer. *Mol. Oncol.* 5, 5–23. doi: 10.1016/j.molonc.2010.11.003

Remy, E., Rebouissou, S., Chaouiya, C., Zinovyev, A., Radvanyi, F., and Calzone, L. (2015). A modeling approach to explain mutually exclusive and co-occurring genetic alterations in bladder tumorigenesis. *Cancer Res.* 75, 4042–4052. doi: 10.1158/0008-5472.CAN-15-0602

Rodriguez, A., Crespo, I., Androsova, G., and del Sol, A. (2015). Discrete logic modelling optimization to contextualize prior knowledge networks using PRUNET. *PLoS oNE* 10:e0127216. doi: 10.1371/journal.pone.0127216

Saadatpour, A., and Albert, R. (2013). Boolean modeling of biological regulatory networks: a methodology tutorial. *Methods* 62, 3–12. doi: 10.1016/j.ymeth.2012.10.012

Saez-Rodriguez, J., Alexopoulos, L. G., Epperlein, J., Samaga, R., Lauffenburger, D. A., Klamt, S., et al. (2009). Discrete logic modelling as a means to link protein signalling networks with functional analysis of mammalian signal transduction. *Mol. Syst. Biol.* 5, 331. doi: 10.1038/msb.2009.87

Stoll, G., Caron, B., Viara, E., Dugourd, A., Zinovyev, A., Naldi, A., et al. (2017). MaBoSS 2.0: an environment for stochastic Boolean modeling. *Bioinformatics* 33, 2226–2228. doi: 10.1093/bioinformatics/btx123

Stoll, G., Viara, E., Barillot, E., and Calzone, L. (2012). Continuous time boolean modeling for biological signaling: application of Gillespie algorithm. *BMC Syst. Biol.* 6:116. doi: 10.1186/1752-0509-6-116

Teschendorff, A. E., Naderi, A., Barbosa-Morais, N. L., and Caldas, C. (2006). PACK: profile analysis using clustering and kurtosis to find molecular classifiers in cancer. *Bioinformatics* 22, 2269–2275. doi: 10.1093/bioinformatics/btl174

Tokheim, C. J., Papadopoulos, N., Kinzler, K. W., Vogelstein, B., and Karchin, R. (2016). Evaluating the evaluation of cancer driver genes. *Proc. Natl. Acad. Sci. U.S.A.* 113, 14330–14335. doi: 10.1073/pnas.1616440113

Wang, J., Wen, S., Symmans, W. F., Pusztai, L., and Coombes, K. R. (2009). The bimodality index: a criterion for discovering and ranking bimodal signatures from cancer gene expression profiling data. *Cancer Inform.* 7:199–216. doi: 10.4137/CIN.S2846

Weinstein, N., Mendoza, L., Gitler, I., and Klapp, J. (2017). A network model to explore the effect of the micro-environment on endothelial cell behavior during angiogenesis. *Front. Physiol.* 8:960. doi: 10.3389/fphys.2017.00960

Zañudo, J. G. T., Scaltriti, M., and Albert, R. (2017). A network modeling approach to elucidate drug resistance mechanisms and predict combinatorial drug treatments in breast cancer. *Cancer Converg.* 1, 5. doi: 10.1186/s41236-017-0007-6

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Advantages of publishing in Frontiers

OPEN ACCESS
Articles are free to read for greatest visibility and readership

FAST PUBLICATION
Around 90 days from submission to decision

HIGH QUALITY PEER-REVIEW
Rigorous, collaborative, and constructive peer-review

TRANSPARENT PEER-REVIEW
Editors and reviewers acknowledged by name on published articles

Frontiers
Avenue du Tribunal-Fédéral 34
1005 Lausanne | Switzerland

Visit us: www.frontiersin.org
Contact us: info@frontiersin.org | +41 21 510 17 00

REPRODUCIBILITY OF RESEARCH
Support open data and methods to enhance research reproducibility

DIGITAL PUBLISHING
Articles designed for optimal readership across devices

FOLLOW US
@frontiersin

IMPACT METRICS
Advanced article metrics track visibility across digital media

EXTENSIVE PROMOTION
Marketing and promotion of impactful research

LOOP RESEARCH NETWORK
Our network increases your article's readership