frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 06 January 2026
DOI 10.3389/fphy.2025.1736037

OPEN ACCESS

EDITED BY
Sauro Succi,
Italian Institute of Technology (IIT), Italy

REVIEWED BY
Sebastiano Pilati,
University of Camerino, Italy
Matthew McFee,
University of Toronto, Canada

*CORRESPONDENCE
Edoardo Milanetti,
edoardo.milanetti@uniroma1.it

RECEIVED 30 October 2025
REVISED 27 November 2025
ACCEPTED 05 December 2025
PUBLISHED 06 January 2026

CITATION
Meta A, Ruocco G and Milanetti E (2026) Neural network–based approach for improving the evaluation of antibody–antigen docking poses.
Front. Phys. 13:1736037.
doi: 10.3389/fphy.2025.1736037

Neural network–based approach for improving the evaluation of antibody–antigen docking poses

Alessandro Meta[1,2], Giancarlo Ruocco[1,2] and Edoardo Milanetti[1,2,3]*

[1]Department of Physics, Sapienza University, Rome, Italy, [2]Center for Life Nano & Neuro Science, Istituto Italiano di Tecnologia, Rome, Italy, [3]Link Campus University, Rome, Italy

The role of artificial intelligence (AI)–based approaches in computational biology and molecular biophysics has become increasingly central over the past decade; however, many challenges remain unresolved, such as the accurate prediction of protein–protein complexes, the complete solution of which would have a significant impact both on our understanding of cellular mechanisms and on the development of therapeutic and diagnostic strategies. Here, we present a protocol based on multiple minimal neural network (NN)–based approaches, trained on a set of carefully selected physicochemical features, to discriminate docking decoy poses (structurally distant from the experimental complex) from native-like poses (structurally close to the native conformation) within a specific class of biologically relevant protein–protein complexes, namely antibody–antigen systems in which the antigen is a protein. A specific version of the proposed method, trained on a set of antibody–antigen interface descriptors, some of which are derived from graph theory to capture the geometric complexity of intermolecular interactions, was compared with ITScore-PP, the docking score provided by HDOCK. This NN-based approach, demonstrates the ability not only to distinguish native-like poses from decoys, but also, more challengly, to discriminate intermediate poses from native-like ones. Furthermore, it was also able to predict the DockQ score, a widely used metric for assessing docking pose quality, showing a larger absolute Pearson correlation coefficient than ITScore-PP. The ability of our NN-based approach, which relies solely on structural interface features, to identify accurate dockings highlights its potential as a valuable tool for improving the ranking of antibody–antigen docking poses and underscores the importance of sppropriate feature selection in protein-protein interaction modeling.

KEYWORDS

AI-driven approaches, antibody–antigen systems, binding modes, binding properties, CDRs, docking scores, decoy docking poses, docking poses

1 Introduction

The field of protein science has experienced a profound transformation in recent years, largely fueled by the rapid development of artificial intelligence (AI) and machine learning approaches [1, 2]. The continuous growth of experimental datasets, together with increasingly sophisticated learning algorithms and advances in high-performance computing infrastructures, especially GPU-based platforms, has led to unprecedented

progress in tackling complex questions in computational biology, bioinformatics, and molecular biophysics [3].

One of the most striking breakthroughs enabled by AI has been the prediction of tertiary protein structures [4]. Algorithms such as AlphaFold2 [5, 6] and RoseTTAFold [7] have fundamentally changed the landscape of structural biology by providing near-experimental accuracy in structural predictions, with a significant impact on protein modeling and rational drug design. Traditional drug discovery is both time-consuming and expensive, but emerging computational methods, including AI-driven approaches, have demonstrated their potential to substantially accelerate the process while reducing costs [8].

Notably, the most significant advances in protein design for therapeutic purposes, including monoclonal antibody engineering, depend not only on accurate single-protein structure prediction but also on the ability to model protein–protein interactions [9–13]. These interactions are central to understanding cellular mechanisms, both physiological and pathological, and are crucial for structure-based drug design strategies. Although the AlphaFold3 algorithm [14] has shown remarkable improvements in predicting biomolecular interactions, further approaches are required to fully exploit both computational power and predictive structural models. In particular, biomolecular binding interfaces display diverse physicochemical properties depending on the molecular partners involved (e.g., protein–protein versus protein–nucleic acid interfaces), highlighting the need for problem-specific feature engineering [15, 16].

Therefore, despite significant progress, predicting the structure of protein–protein complexes remains a challenging task, particularly in the case of antibody–antigen systems [17, 18], which are extensively studied due to their importance in both therapeutic and diagnostic applications. AI-based methods offer unique advantages in this context [19, 20], providing data-driven strategies that can complement physics-based approaches and capture subtle structural patterns associated with molecular recognition.

Over the past decade, antibodies have emerged as powerful therapeutic agents, benefiting from technological advances that allow their structure and function to be characterized with increasing precision. Effective antibody design requires a deep understanding of the structural determinants of antibody–antigen recognition. While experimental methods such as X-ray crystallography, cryo-electron microscopy, NMR, and mutagenesis provide high-resolution insights, they are resource-intensive and time-demanding. Computational approaches, particularly molecular docking, represent a valuable and efficient alternative. Several docking platforms, including ClusPro [21], LightDock [22], ZDOCK [23], HDOCK [24], and HADDOCK [25], have been developed to generate docking poses of antibody–antigen complexes [26, 27]. However, identifying near-native conformations remains challenging, as current scoring functions are often optimized for binding affinity rather than structural accuracy. Deep learning methods are increasingly being explored to overcome these limitations by directly extracting informative patterns from structural data [28, 29]. In this context, we present a study emphasizing the role of careful feature selection and combination strategies in describing antibody–antigen interfaces for predictive modeling using both supervised and unsupervised machine learning methods.

Here, we explore the application of minimal yet effective machine learning (ML) techniques, in particular using Neural Network (NN), to the analysis and discrimination of docking poses in antibody–antigen complexes. We take into account both supervised and unsupervised approaches, considering in particular the principal component analysis (PCA), to evaluate their ability to distinguish between native-like and fully decoy docking conformations. Furthermore, we demonstrate that a simple NN trained on a set of interface descriptors, some of which are derived from graph-theoretical representations, can not only separate native-like from decoy poses but also correlate strongly with DockQ score [30], a widely used metric for evaluating docking quality (which is defined as a linear combination of rescaled CAPRI-standard evaluation metrics [31] (see Equations 1–3). Finally, we compare the performance of this minimal NN-based framework with the docking score produced by HDOCK, which has already been used to test the predictive capability of antibody–antigen structural models [26], highlighting its potential as a complementary strategy to improve the ranking of antibody–antigen docking poses. In this context, the choice of the docking method is not central, since the methodological requirement is solely to generate both native-like and decoy docking poses, which serve as the basis for training the predictive algorithm, regardless of the success rate of the docking method employed. More specifically, we analyze a dataset of approximately 2,200 experimentally resolved antibody–antigen complexes. For each complex, docking was performed using HDOCK to generate a pool of docking poses, which were then classified as decoys or native-like according to the DockQ score. Overall, the presented approach demonstrates how feature engineering combined with AI-driven approaches can effectively classify and predict the quality descriptor of docking poses of antibody–antigen conformations, thereby supporting future developments in structure-based antibody design.

2 Results

Despite the significant progress that ML techniques have brought to the field of computational biology, improving the evaluation of docking poses remains a challenge that is not yet fully solved [32–35]. Here, we show that the appropriate selection of features capable of capturing the geometric properties of the interface between predicted dimeric structures, when used in simple NN models, can help improve the assessment of docking poses provided by the docking score.

In particular, we employed a set of antibody–antigen complexes (considering only protein antigens), since incorrectly predicted poses may involve regions other than the complementarity-determining regions (CDRs), which consist of six hypervariable loops and exhibit physicochemical properties that differ considerably from those of the native conformations.

This work focuses on discussing the selection of the number of parameters in a simple NN to achieve generalizable discrimination between decoy and native-like docking poses, as well as accurate prediction of the DockQ score, which is typically used to evaluate the quality of a docking pose. The results are discussed in the following sections.

2.1 Dataset analysis and definition of native-like and decoy docking poses

As a first analysis, we investigated the composition of the dataset used, focusing in particular on the docking poses classified as well-predicted (i.e., structurally similar to the experimentally resolved native conformation of the complex) and on those incorrectly predicted (i.e., with structures that are considerably different from the corresponding experimental native conformation). To this end, we employed the DockQ score (see Methods for a more detailed description), which is able to classify native-like poses and decoy poses according to a threshold value.

In Figure 1a, we report the DockQ distribution for all docking poses generated by the HDOCK method. In particular, for each antibody–antigen docking simulation, we considered the top 10 docking poses ranked by score. The distribution shows that the two main peaks of the probability density function (PDF) are centered at low DockQ values, which are less then 0.24, and at high DockQ values, which are higher than 0.81, indicating that only a small fraction of poses are predicted as native-like (DockQ > 0.81), while the majority correspond to decoy poses (DockQ < 0.24). A cartoon representation of the structural alignment between the docking pose and the native structure, for different ranges of DockQ values, is shown in Figure 1c. This clearly highlights the difficulty of docking algorithms in accurately predicting the native conformation of interacting proteins. Very high DockQ values (close to 1) typically correspond to very small Root Mean Square Deviation (RMSD) values, which can be interpreted as structural fluctuations within thermal noise of experimentally determined native conformations [36, 37]. Therefore, the ability of the approaches proposed in the following sections must rely on identifying, based on specific interfacial structural properties, the decoy docking poses. In this way, the algorithm can be trained to discriminate between decoy and native-like conformations in a fully general manner, even when the predicted antibody–antigen complex exhibits an interface significantly different from those included in the training set.

In particular, according to the DockQ values calculated for each docking pose, the overall dataset is composed as follows: 19,406 decoy docking poses, 790 intermediate docking poses, and 1,684 native-like docking poses (see Figure 1a; Table 1). As shown by the bimodal trend in the distribution in Figure 1a, in most cases the docking method returns a pose that is structurally distant from the reference structure (i.e., the experimentally resolved complex). However, for 59% of the complexes in the dataset, the top-ranked pose generated by the algorithm is classified as native-like, in some cases with a very high DockQ score, making the docking model and the experimentally determined native structure nearly indistinguishable. This behavior may be due to the algorithm's prior knowledge of the native structure (or its homologs), as well as particularly easy cases for the algorithm to predict. Nevertheless, this does not hinder the aims of the present work, which first seeks to classify docking poses according to their DockQ value and subsequently to predict the descriptor. In light of this, the development of computational methods capable of identifying decoy docking poses is crucial, as it helps reduce the space of possible binding conformations (by removing these from the candidate solutions) that require further investigation. A deeper insight on the overall dataset is presented in Table 1.

In this study, we defined two different subsets. The first, referred to as the Decoy–Native-like dataset (DNL dataset), includes only decoy and native-like docking poses and is used for the classification approach. The second, which comprises all three classes (decoy, intermediate, and native-like) and is referred to as the DINL dataset, is used to predict the DockQ value of a generic docking pose. The DNL dataset consists of 1,587 decoy docking poses and an equal number of native-like docking poses. Conversely, the DINL dataset, in which the DockQ value of each docking pose is taken into account, is composed of 1,000 decoy poses, 790 intermediate poses, and 1,000 native-like docking poses (see Figure 1a).

Each docking pose is characterized by a docking score, ITScore-PP [38], which is a numerical value used to rank the predicted binding modes of molecules—more negative scores indicate more stable and likely interactions. The distribution of ITScore-PP values is shown in Figure 1b for the native-like and decoy groups separately. The difference between the two distributions is evident, and the classification based on the ITscore-PP descriptor provided by the HDOCK docking method yields an Area Under the Receiver Operating Characteristic (ROC) Curve (AUC) of 0.78 (Equation 24).

The aim of this work is to investigate how a minimal neural network–based approach can improve the classification of docking poses into native-like and decoy categories when an appropriate selection of binding properties is adopted, and how it can directly predict the DockQ value on which this classification is based.

2.2 Correlation analysis among the features

To evaluate the docking poses, an initial set of 21 features was defined and is listed and described in Table 2 (see Equations 4–19 for further details). The selected features primarily describe geometric properties at the interface between antibody–antigen docking poses, some of which are based on graph theory to better capture the complexity of the geometric organization of the residues involved in intermolecular interactions.

An initial Pearson correlation analysis was performed to remove pairs of features showing high correlation (absolute Pearson correlation coefficient > 0.75, for both positive and negative correlations). The correlation matrices for all pairs of features, both for the initial 21 features and for the 15 features remaining after filtering, are shown in Figure 2a. In particular, in order to remove highly correlated feature pairs while minimizing feature pruning, the absolute Pearson correlation coefficients were mapped onto a graph in which pairs of strongly correlated features were connected. The resulting problem is equivalent to a minimum vertex cover problem, which was solved exactly using integer linear programming (ILP), given the small number of highly correlated features (see Equation 20). In addition, for both matrices, the corresponding graphs are displayed, where each node represents a feature and each edge between two features is weighted (using a red-to-blue color scale) according to their Pearson correlation.

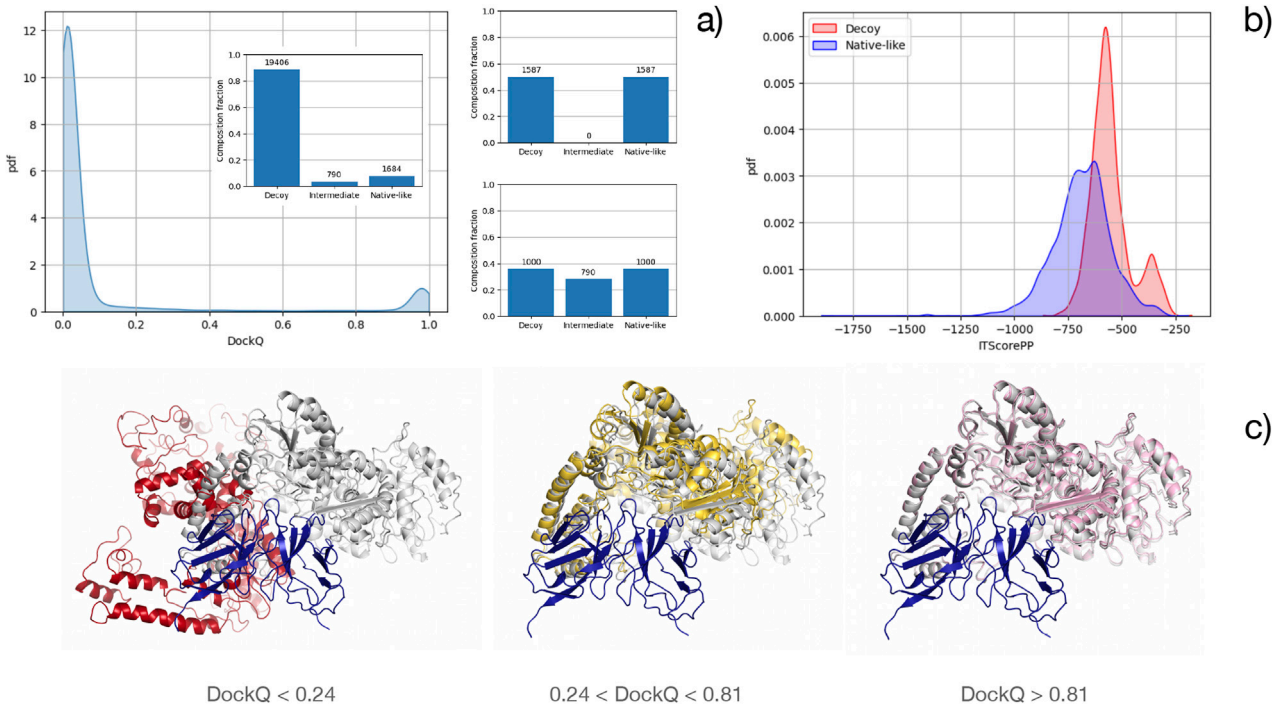

FIGURE 1
Docking pose dataset analysis and class definition. **(a)** Probability density function of DockQ values for all docking poses in the dataset. The inset shows the fraction (and absolute number) of poses in the three classes: Decoy (DockQ < 0.24), Intermediate (0.24 < DockQ < 0.81), and Native-like (DockQ > 0.81). On the right, the class proportions (and absolute numbers) are reported for the DNL and DINL datasets, respectively. **(b)** Probability density function of the docking score (ITScore-PP) provided by the HDOCK method. Red and blue curves represent the distributions for the decoy and native-like classes, respectively. **(c)** Cartoon representation (example) of antibody–antigen docking structures across different DockQ ranges. The experimentally resolved antibody and antigen structures are shown in blue and gray, respectively, while the antigen structures placed by the docking algorithm are shown in red (decoy, DockQ < 0.24), yellow (intermediate, 0.24 < DockQ < 0.81), and pink (native-like, DockQ > 0.81).

2.3 Unsupervised classification of native-like and decoy docking poses through PCA

The selection of 15 largely independent features, after appropriate normalization (see Methods), allowed us to perform a Principal Component Analysis (PCA). For each docking pose in the DNL dataset, which is used to classify docking poses as decoy or native-like, a vector of 15 normalized features was associated. The PCA results are shown in Figure 2b, and the proportion of variance explained by each eigenvector is reported in Figure 2c, showing that the first two principal components account for 32% of the total variance. The unsupervised PCA approach was employed here to explore a potential blind classification of the two docking pose classes (decoy and native-like). Each point in Figure 2b represents the projection of the 15-dimensional feature vector (associated with a single docking pose) onto the essential plane defined by the first two principal components (PC1–PC2). Points are colored red and blue according to their membership in the decoy or native-like class, respectively.

The analysis of the two distributions (decoy and native-like) along PC1 does not reveal a clear separation between the two classes, as evidenced by the strong overlap between distributions. This is also confirmed by the ROC curve shown in Figure 2b, with an AUC of 0.52, which is effectively close to random classification. By contrast, the distributions of decoy and native-like poses along PC2 are noticeably more separated, yielding an ROC AUC of 0.68 (see Figure 2e). Therefore, the use of PC2 alone, in a fully unsupervised manner, provides a moderate but non-negligible discriminative power between decoy and native-like classes.

The loading analysis in this context reveals the contribution of each feature to the definition of each principal component. In particular, the interface properties most relevant for the separation between decoy and native-like poses are pca_flatten_ratio, pca_alignment_score, pca_stretch_ratio_bs and pca_flatten_ratio_bs (features 2, 3, 9 and 10, see Table 2), which show a more pronounced difference compared to the corresponding loadings of PC1. The first two features are related to the geometry of the antibody–antigen interaction. Specifically, the first feature reflects the globularity of the complex, which increases when the interface lies in proximity to the CDR, while the second describes the relative orientation of the two molecules. Instead, the last two features capture the circularity and concavity of the binding interface, with higher PC2 values corresponding to a flatter interaction surface. A comparison between the ROC curves of PC1 and PC2 (with ROC AUCs of 0.52 and 0.68, respectively) and that of the HDOCK docking score (ROC AUC of 0.80, which was calculated using the DNL dataset) is reported in Figure 2e, highlighting the need to develop supervised methods to better evaluate each docking pose based on interfacial geometric properties.

TABLE 1 Overall dataset distribution.

All the docking poses	
Decoy docking poses	88.69%
Intermediate docking poses	3.61%
Native-like docking poses	7.70%
All the native complexes	
Complexes with at least one decoy docking pose	99.95%
Complexes with at least one intermediate docking pose	20.02%
Complexes with at least one native-like docking pose	73.17%
The top-ranked docking poses	
Top-ranked best docking poses	57.54%
Native-like top-ranked docking poses	59.28%

The second column indicates the percentage of the class presented in the first column. For the first group of classes the frequency has to be intended among all the docking poses, for the second group among all the native complexes, while for the third one among the top-ranked docking poses.

2.4 A minimal neural network–based approach to classify native-like and decoy docking poses

In this section, a minimal neural network (see Methods and Equations 21–23) is employed with the aim of improving the classification between native-like and decoy docking poses as provided by the docking score. The goal is to investigate the contribution of neural network–based approaches to enhancing docking pose evaluation. As a first step, the training and test sets were randomly selected. Subsequently, in order to make the procedure as general as possible, multiple training and test datasets were generated so as to be maximally distinct with respect to the features selected for this study.

The first approach was therefore performed by considering one training set and multiple test groups, both drawn from the DNL dataset (see Methods for further details). The predictive performance, in terms of the Area Under the ROC Curve (AUC), was analyzed as a function of the number of network parameters, varying both the total amount of available data and the ratio between training and test samples. The results of this preliminary analysis are shown in Figure 3a, which highlights that, for a number of parameters equal to 70, a plateau in the test AUC curve (in red) is observed in almost all cases considered. By selecting the total number of available complexes—after verifying that this amount is sufficient to capture the information required to discriminate between the two classes—together with a 0.5 ratio between training and test data and a total of 70 network parameters, we obtained an average test set ROC AUC value of 0.90. This value exceeds the ROC AUC calculated using only the docking score. The comparison between the corresponding ROC curves is also reported in Figure 3c. An analogous analysis performed on the discrimination between native-like and intermediate docking poses in the DINL dataset yields a ROC AUC of 0.77, indicating a promising classification performance on this substantially more challenging and subtle task, where the structural differences between the pose classes are markedly smaller than in the native-like vs. decoy scenario.

The importance of the selected descriptors, as indicated by the PC2 loadings, can be assessed by training the NN after removing the descriptors with loadings greater than 0.20 (i.e., those contributing most to the separation between decoy and native-like docking poses according to PC2). This procedure results in a classification performance on the test set that is 14% lower than the performance obtained when retaining all selected features.

In addition, we propose training the NN on training and test sets that are as different as possible in terms of the selected features, in order to make the NN-based classification procedure as generalizable as possible. To this end, the entire dataset was split into two parts (training and test sets, and then swapped) according to the value of PC1, i.e., the projection of the feature vector onto the first principal component of each docking pose. Docking poses with PC1 values below the mean were assigned to one group, while those with higher PC1 values were assigned to the other.

The choice of PC1 as the reference distribution for defining the two groups was motivated by two considerations: (i) PC1, by definition, is the eigenvector associated with the largest proportion of explained variance, thus carrying the highest amount of information; and (ii) the distributions of the PC1 values for the decoy and native-like docking poses show no clear separation (and therefore no intrinsic discriminative power between the two classes), unlike PC2 (see Figure 2). This ensures that, before and after splitting by the PC1 mean, the relative proportion of decoy and native-like poses within each subset remains approximately the same, see Figure 3b.

The results are shown in Figure 3c, which illustrates a neural network discriminative capability between decoy and native-like docking poses that is intermediate between the ITScore-PP docking score provided by HDOCK (ROC AUC of 0.80) and the NN previously trained on randomly selected training and test subsets. A ROC AUC of 0.90 was measured for the NN trained and tested on randomly selected sets, decreasing to values that span between 0.81 and 0.82, when the training and test sets are separated based on the PC1 values associated with each docking pose. In particular, the improved classification capability of the proposed NN-based approach is further confirmed by the steep initial rise of the ROC curves corresponding to the NN-based methods, observed in the early phase (at low true positive rate and false positive rate values). This result highlights the ability of an NN-based approach, when coupled with properly selected features, to improve docking classification performance even when the training and test sets are deliberately constructed to have different underlying properties.

2.5 The use of neural networks to improve the evaluation of docking poses

In the previous section, we demonstrated the importance of employing simple NN models for the classification of docking

10.3389/fphy.2025.1736037

TABLE 2 Table of features (see methods for a more detailed description).

1	pca_stretch_ratio	Ratio between the values of the first and second components of the explained variance of a PCA performed on the residues coordinates. It represents the stretching of the complex shape
2	pca_flatten_ratio	Ratio between the values of the second and third components of the explained variance of a PCA performed on the residues coordinates. It represents the flatness of the complex shape
3	pca_alignment_score	Absolute cosine of the angle between the main principal components of a PCA performed both on the antibody and the antigen residues coordinates
4	bs_sasa_ratio	The fraction of the complex SASA (solvent-accessible surface area) involved in the binding sites
5	bs_size	Number of residues in the binding site
6	ab_bs_size	Number of residues in the antibody binding site
7	ag_bs_size	Number of residues in the antigen binding site
8	pca_normalized_centroid_distance	Distance between the centroids of the antibody and the antigen, normalized through the main principal component of the PCA performed on the coordinates of the whole complex residues
9	pca_stretch_ratio_bs	Equivalent to feature 1 for the binding sites residues
10	pca_flatten_ratio_bs	Equivalent to feature 2 for the binding sites residues
11	bs_mean_hydrophobicity	Average hydrophobicity of the binding sites residues
12	bs_delta_hydrophobicity	Absolute difference of average hydrophobicity between the antibody and the antigen binding sites
13	edge_density	Edge density of the unweighted network composed by the complex residues interactions
14	mean_degree	Average degree of the unweighted network
15	mean_strength	Average strength of the weighted network ($w(A,B) = 1/dist(A,B)$ for any couple of interacting residues)
16	network_diameter	Diameter of the weighted network
17	network_radius	Radius of the weighted network
18	mean_assortativity	Average degree assortativity of the networks
19	unweighted_mean_clustering	Average clustering coefficient of the unweighted network
20	weighted_mean_clustering	Average clustering coefficient of the weighted network
21	network_transitivity	Transitivity of the networks

poses into decoy and native-like categories in antibody–antigen complexes. Furthermore, we emphasized the crucial role of accurate feature selection, which, combined with supervised machine learning methods, can significantly improve predictive performance. Here, a minimal feedforward neural network is trained to directly correlate (rather than classify) with the DockQ value, which is one of the standard metrics used to assess the quality of a docking pose. For this purpose, the DINL dataset was taken into account (see Methods for further details).

In this case as well, the scatterplot of the first two principal components obtained from the PCA of the feature vectors of all docking poses is shown in Figure 4a, where each point (docking pose) is colored according to its corresponding DockQ value. Given the inherent difficulty of capturing, through unsupervised approaches such as PCA, the relationship between the interface descriptors of predicted antibody–antigen complexes and their structural deviation from the corresponding experimentally resolved native structures (quantified by DockQ), we developed a neural network (NN) model trained on the DINL dataset.

By randomly selecting the training and test sets (see Methods for further details), we statistically analyzed how the Pearson correlation (Equation 25) in the test set between the experimental DockQ and the predicted DockQ (pDockQ) varies as a function of the number of NN parameters. The results for a NN trained on 80% of the available poses, reported in Figure 4a, show that a substantial performance gain is achieved by increasing the number of parameters up to approximately 70. Beyond this point, the correlation between DockQ and pDockQ increases much more slowly, while the mean square error (MSE) on the training set reaches a plateau for ≈86,400 parameters (Figure 4a).

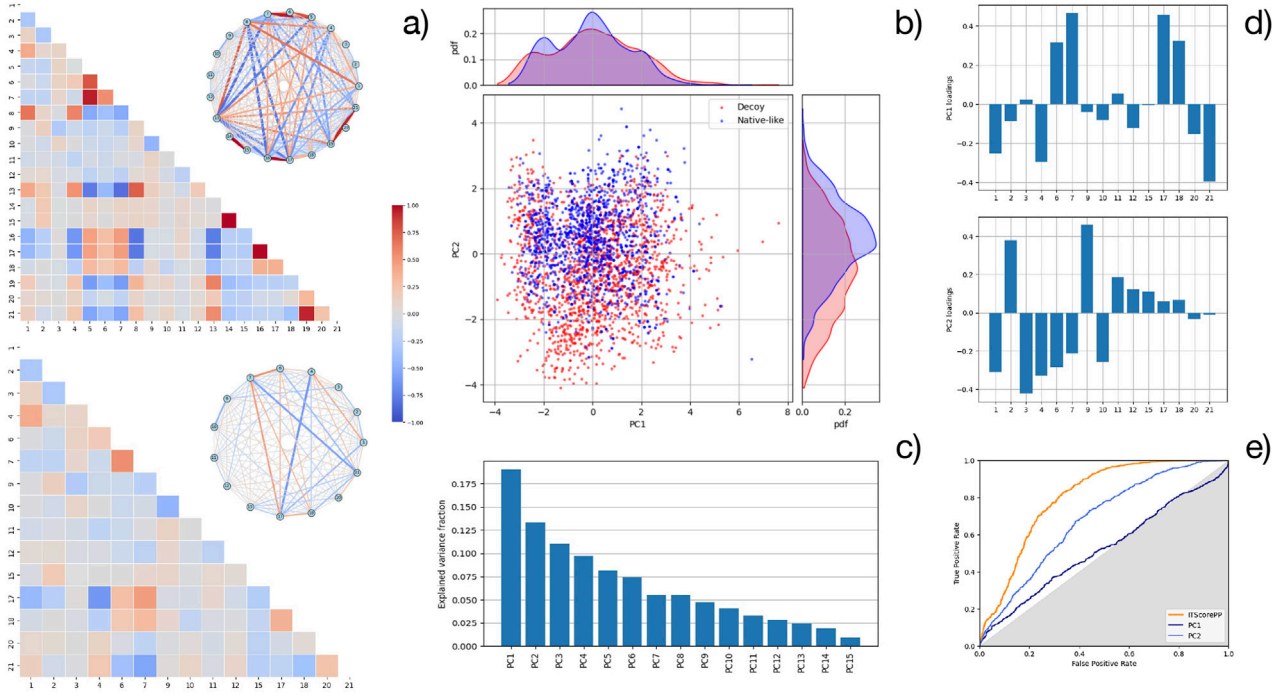

FIGURE 2
Correlation analysis and PCA. **(a)** Pearson correlation matrices for all initial features (top) and for the selected features after removing highly correlated pairs (bottom). For each matrix, a fully connected graph is shown, where each node represents a feature and each edge is weighted by the corresponding Pearson correlation value. **(b)** Projection of the 15-dimensional feature vectors onto the essential plane defined by PC1 and PC2. Red and blue points represent decoy and native-like docking poses, respectively. Probability density functions for PC2 (top) and PC1 (right) are shown. **(c)** Explained variance ratio for each of the 15 principal components. **(d)** Feature loadings for PC1 (top) and PC2 (bottom). **(e)** ROC curves for the decoy vs. native-like distributions along PC1 (blue), PC2 (cyan), and for the docking score ITScore-PP (orange).

Furthermore, the variation in correlation between pDockQ (computed with an 86,400 parameters NN) and DockQ has been studied as a function of the training set size, spanning from 50% of the DINL dataset to 93%, alongside the difference in MSE (ΔMSE) between training and test set. For both the measures, the results, reported in Figure 4b, show an optimal average value for the 80% training set proportion.

For the NN trained with 86,400 parameters, on a set composed by 80% of the docking poses, the correlation between DockQ and pDockQ in the DINL set is 0.59.

The comparison between neural network predictions (pDockQ) and the docking score (ITScore-PP) was performed by evaluating the correlation with the DockQ score. In addition, we used projections onto the first two principal components (PC1 and PC2) of each docking score as potential predictors of DockQ. The scatter plots showing the relationship between DockQ and each proposed predictor (supervised and unsupervised) are reported in Figure 4c.

In particular, the correlations between DockQ and PC1, PC2, ITScore-PP (the docking score), and pDockQ are −0.04, −0.27, −0.41, 0.59 respectively (Table 3). This indicates that the NN-based approach, which uses as input the 15 selected features, substantially improves the quantitative evaluation of docking poses compared to the original docking score. Of particular note is the correlation between DockQ and the second principal component (PC2) of the PCA performed on the features. As a fully unsupervised descriptor, PC2 provides insight into the features that contribute most to the definition of the component (loadings), thereby offering the opportunity to further refine NN-based models through preliminary feature selection procedures.

To better illustrate the ability of the neural network–based predictive method to estimate DockQ values even in intermediate cases ($0.24 < \text{DockQ} < 0.81$), a probability density function (PDF) was computed for each DockQ range (analogous to the boxplot analysis shown in Figure 4c). The distributions of the pDockQ descriptor are progressively shifted with increasing DockQ ranges (see Figure 4d), thereby confirming the method's ability not only to classify docking poses as native-like or decoy—as also supported by this DockQ estimation procedure—but also to correlate with intermediate DockQ values, with slightly lower yet satisfactory accuracy compared to the docking score.

Furthermore, to assess the overall quality of the pDockQ descriptor, it has been benchmarked in terms of Pearson correlation coefficient against both ITScore-PP and the predicted binding free energy (ΔG), obtained using an MM/GBSA-based predictor [39] via the HawkDock server [40]. This comparison has been performed on a randomly selected small subset of the DINL dataset composed by 84 docking poses (30 decoy, 24 intermediate and 30 native-like poses, in order to maintain the proportions of the DINL dataset). For this analysis, the pDockQ values have been computed by a 86,400 parameters NN trained on all the DINL docking poses that do not

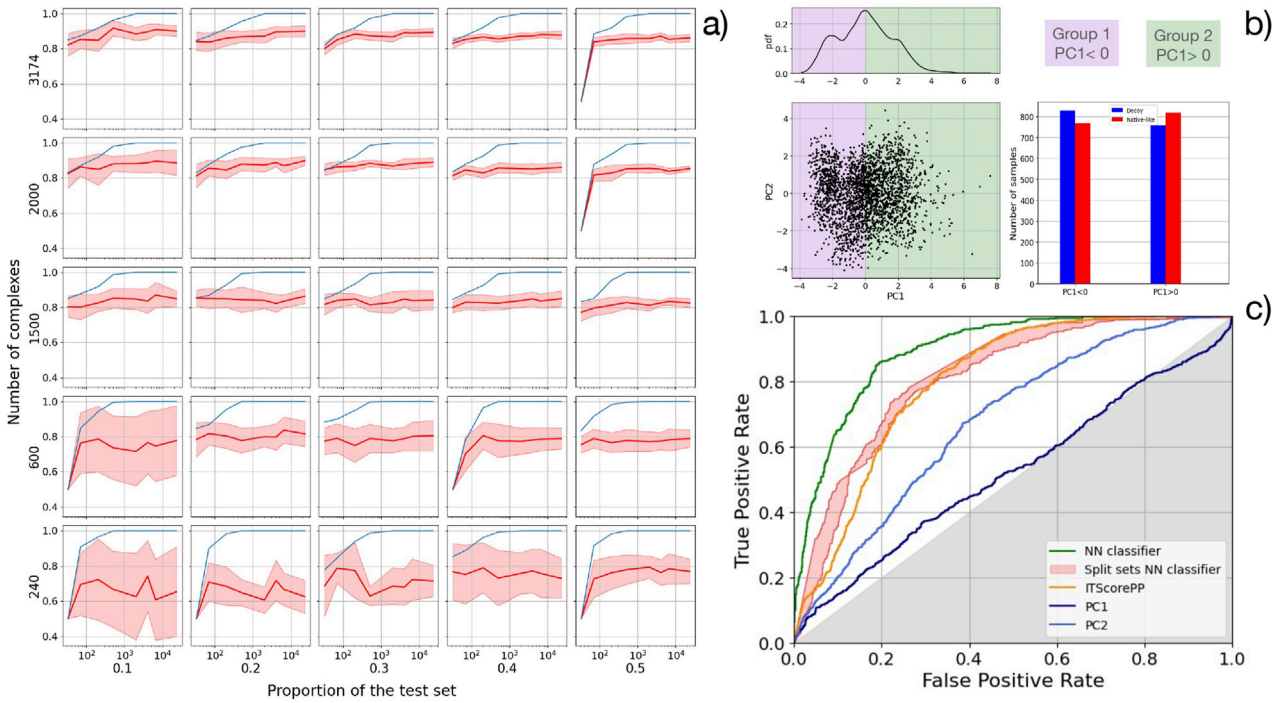

FIGURE 3
Performance of the neural network (NN) in docking pose classification. **(a)** Each plot shows the ROC AUC as a function of the number of parameters used in the NN for the DNL dataset. From left to right, the test set proportion increases, while from bottom to top, the number of complexes used increases. **(b)** The scatterplot displays the first two principal components (PC1 and PC2) obtained from the PCA of normalized feature vectors, along with the probability density function of PC1. Values below the mean (zero) are colored in purple, while those above the mean are colored in green. On the right, the number of complexes classified as decoy and native-like are reported for the first (PC1 < 0) and second (PC1 > 0) groups, respectively. **(c)** ROC curves are shown for classifications based on the first two principal components of the features (blue and light blue for PC1 and PC2, respectively), the docking score (ITScore-PP, orange), the NN trained and tested on randomly selected sets (green), and the NN trained and tested on sets defined according to differences in docking poses along PC1 (red).

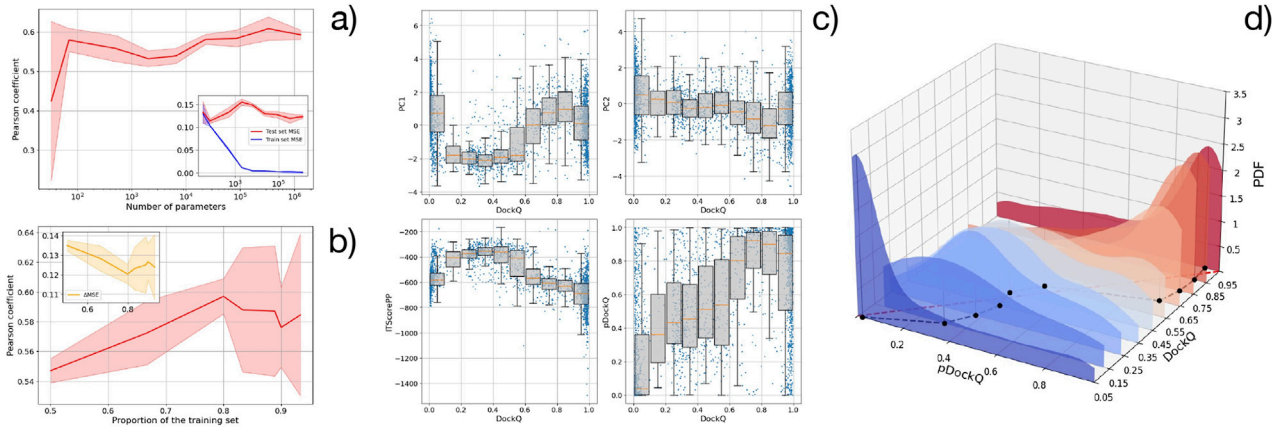

FIGURE 4
Performance of the neural network (NN) in DockQ prediction. **(a)** The Pearson correlation coefficient between the predicted DockQ (pDockQ) and the measured DockQ is shown as a function of the number of parameters used to train the NN across different test sets from the DINL dataset. The inset reports the trend of the mean squared error (MSE) for both training and test sets as a function of the number of parameters. **(b)** The Pearson correlation coefficient between the predicted DockQ (pDockQ) and the measured DockQ is shown as a function of the training set size used to train the NN across different test sets from the DINL dataset. The inset reports the difference in mean squared error (ΔMSE) between training and test sets as a function of the training set size. **(c)** For each panel, the scatterplot (blue points) is combined with a boxplot (in gray) of the measured DockQ, together with the four descriptors: PC1, PC2, docking score (ITScore-PP), and the NN-predicted DockQ (pDockQ). **(d)** Probability density function (PDF) of pDockQ for different DockQ ranges.

TABLE 3 Results recap. The reported correlation value refers to the Pearson correlation coefficient.

Descriptor	DNL ROC AUC	Split sets ROC AUC	Corr. with DockQ (p-value)
PC1	0.52	0.54	−0.04 (0.03)
PC2	0.68	0.62	−0.27 ($< 10^{-7}$)
ITScore-PP	0.80	0.84	−0.41 ($< 10^{-7}$)
pDockQ	0.90	0.81	0.59 ($< 10^{-7}$)

The p-value refers to the null hypothesis that the distributions underlying the samples are uncorrelated and normally distributed.

share the reference native complex with any test set element. While ΔG and pDockQ show a comparable performance (respectively −0.62 and 0.67), both have a significantly larger Pearson correlation coefficient in magnitude than ITScore-PP (−0.43). Although the small size of the test set of this assessment does not allow a definitive statement, the pDockQ approach results are promising, even when compared with one of the state-of-the-art methods reported in the literature.

3 Conclusion

In this work, we addressed the role of minimal neural networks (NNs) in tackling the still unresolved problem of accurately evaluating docking poses. Specifically, for a set of experimentally determined antibody–antigen complexes, structural predictions were generated using the HDOCK docking method. Each predicted pose is associated with a docking score that is intended to reflect its reliability based on an internal scoring function. The main idea of this study is to improve the assessment of docking scores through the use of neural networks. Each docking pose was evaluated by structural comparison with the experimentally resolved native complex using the DockQ metric, which is commonly employed to assess the performance of molecular docking prediction methods. Threshold values of DockQ were then used to classify docking poses as decoy or native-like. A set of physicochemical features, some of which are derived from graph theory to capture the complexity of residue–residue interactions at the antibody–antigen interface, was defined with the aim of training one NN for the classification between decoy and native-like poses, and another NN for the direct prediction of DockQ. The results show that, unlike the unsupervised descriptors obtained from the principal components (PCA) of normalized features, the two trained NNs significantly improved both the classification between native-like and decoy poses, as well as between intermediate and native-like docking poses, and the direct prediction of DockQ compared to the docking score provided by HDOCK. These findings highlight the importance of neural network–based approaches, combined with the selection of chemically and physically relevant features, in improving the evaluation of docking poses and in describing antibody–antigen binding interactions.

4 Methods

4.1 Dataset of antibody-antigen complexes

The initial dataset consisted of 9,780 experimentally resolved antibody–antigen complexes retrieved from the SAbDab database [41]. A first filtering step was applied to retain only complexes in which the antigen was classified as a protein or peptide and consisted of a single chain (thus preserving only monomeric antigens), resulting in 9,486 structures. Structures containing missing residues were either repaired or removed from the dataset, yielding 9,463 structures.

A multiple sequence alignment among all antibody–antigen complexes in the dataset was performed to remove redundancy. Specifically, for each complex we considered a single sequence obtained by concatenating the antibody sequence (heavy and light chains) with the sequence of the corresponding antigen. These sequences were then processed with CD-HIT [42–44] using a sequence identity cutoff of 0.9, resulting in 2,517 centroids, which represent the most representative sequences in the entire dataset. Since the study focuses on the calculation of interface properties, it was crucial to ensure that the interfaces were complete, i.e., without missing residues in the binding region. Therefore, complexes with incomplete interfaces were excluded, reducing the dataset from 2,517 to 2,244 structures.

Finally, energy minimization was performed on all structures, resulting in a final set of 2,188 properly minimized complexes.

4.2 Docking simulation of antibody–antigen complexes and decoy pose selection

Each antibody–antigen complex with a known experimental structure was split into two separate structures, antibody and antigen, which were then subjected to molecular docking simulations using the HDOCK method (thus considering the interacting structures in their bound conformations). For each antibody–antigen docking simulation, the top ten poses proposed by the method were retained. Each docking pose was evaluated using the DockQ metric, which is defined according to the following formula:

$$DockQ = \frac{F_{nat} + lRMS_{scaled} + iRMS_{scaled}}{3} \tag{1}$$

with

$$lRMS_{scaled} = \frac{1}{1 + \left(\frac{lRMS}{8.5\text{Å}}\right)^2} \tag{2}$$

and

$$iRMS_{scaled} = \frac{1}{1 + \left(\frac{iRMS}{1.5\text{Å}}\right)^2}, \tag{3}$$

where F_{nat}, $lRMS$ and $iRMS$ are the CAPRI-standard classification metrics [30, 31].

In particular, for the DNL dataset, we selected for each experimental complex the "decoy" docking pose as the one associated with the lowest DockQ value among the ten poses

considered (ensuring in all cases that DockQ < 0.24), and identified the "native-like" pose as the one with the highest DockQ value among the ten poses generated by HDOCK (with DockQ > 0.81). Furthermore, 1,000 docking poses classified as decoy (based on their DockQ values) and 1,000 docking poses classified as native-like (also based on DockQ) were randomly selected from the docking poses obtained after the previous filtering steps. These poses were used to define the DINL dataset, which served as the training set for the neural network designed to predict DockQ values.

4.3 Feature description

The features used throughout the whole paper can be divided into three groups: complex geometry features, interface features, complex graph features. The first group comprises all the measures related to the geometrical arrangement of the antibody-antigen complex α-carbon atoms. The first group is composed by

- pca_stretch_ratio:

$$l_1 = \frac{\lambda_2}{\lambda_1}, \tag{4}$$

where λ_1, λ_2 and λ_3 are the first, second and third component of the explained variance of a PCA performed on the coordinates related to the α-carbons of the whole complex;

- pca_flatten_ratio:

$$l_2 = \frac{\lambda_3}{\lambda_2}; \tag{5}$$

- pca_alignment_score:

$$\theta = |\hat{v}_1^{ab} \cdot \hat{v}_1^{ag}|, \tag{6}$$

where $\hat{v}_1^{ab}$ and $\hat{v}_1^{ab}$ are the two unit vectors on the direction of the main principal component of the PCAs performed separately on the antibody and the antigen;

- pca_normalized_centroid_distance:

$$CD = \frac{dist\left(C_{ab}, C_{ag}\right)}{\lambda_1}, \tag{7}$$

where $dist(C_{ab}, C_{ag})$ is the distance between the centroids of the antibody and the antigen.

The second group is composed by features accounting for several properties of the antibody-antigen binding site (BS). For this group, we defined as BS residues those residues whose α-carbon is within $12\mathring{A}$ to an α-carbon atom from a different molecule (i.e. the antibody residues closer than $12\mathring{A}$ to an antigen residue and vice versa). The second group features are

- bs_sasa_ratio:

$$\frac{SASA^{BS}}{SASA_{tot}} = \frac{\mathrm{SASA}_{ab}^{BS} + \mathrm{SASA}_{ag}^{BS}}{\mathrm{SASA}_{ab} + \mathrm{SASA}_{ag}}, \tag{8}$$

where SASA_{ab} and SASA_{ag} are the solvent-accessible surface area (SASA), respectively, of the unbound antibody and antigen, while SASA_{ab}^{BS} and SASA_{ag}^{BS} represent the SASA values of the corresponding unbound antibody and antigen BS residues;

- bs_size: number of BS residues;
- ab_bs_size: number of antibody BS residues;
- ag_bs_size: number of antigen BS residues;
- pca_stretch_ratio_bs:

$$l_1^{BS} = \frac{\lambda_2^{BS}}{\lambda_1^{BS}}, \tag{9}$$

where λ_1^{BS}, λ_2^{BS} and λ_3^{BS} are the first, second and third component of the explained variance of a PCA performed on the coordinates related to the α-carbons of the BS residues;

- pca_flatten_ratio_bs:

$$l_2^{BS} = \frac{\lambda_3^{BS}}{\lambda_2^{BS}}; \tag{10}$$

- bs_mean_hydrophobicity: average hydrophobicity of the BS residues, according to the water orientation probability hydropathy scale (WOPHS) [45];
- bs_delta_hydrophobicity: absolute difference in average hydrophobicity (according to the WOPHS) between the antibody and antigen BS residues.

The SASA values are measured using the Shrake-Rupley "rolling ball" algorithm with probe radius of $1.40\mathring{A}$ and definition of 100 points/$\mathring{A}^2$ via the *Biophyton* library [46].

The third group features are common graph theory descriptors measured on two networks: an unweighted network, where all nodes corresponding to residues whose α-carbons are within $12\mathring{A}$ are linked, and a weighted network, where to any edge (i,j) of the unweighted network is assigned a weight $W_{ij} = 1/dist(i,j)$. The following features belong to the third group:

- edge_density: edge density of the unweighted network

$$\rho = \frac{1}{N(N-1)} \sum_{i,j}^{1,N} L_{ij}, \tag{11}$$

where N is the number of nodes, L is the adjacency matrix, i.e. $L_{ij} = 1$ if i and j are connected, 0 otherwise, and $k_i = \sum_{j=1}^{N} L_{ij}$ is the node i degree.

- mean_degree: average degree of the unweighted network

$$k = \frac{1}{N} \sum_{i=1}^{N} k_i = \frac{1}{N} \sum_{i,j}^{1,N} L_{ij}; \tag{12}$$

- mean_strength: average strength of the weighted network

$$s = \frac{1}{N}\sum_{i,j}^{1,N} W_{ij}; \tag{13}$$

- network_diameter: diameter of the weighted network

$$d = \max_j \left\{\max_i \left\{e_{ij}\right\}\right\}, \tag{14}$$

where e_{ij} is the length of the shortest path between nodes i and j on the weighted network and it is measured via Dijkstra's algorithm;

- network_radius: radius of the weighted network

$$r = \min_j \left\{\max_i \left\{e_{ij}\right\}\right\}; \tag{15}$$

- mean_assortativity: average degree assortativity of the networks

$$a = \frac{\sum_{n,m} nm\left(f(n,m) - q_n q_m\right)}{\sigma_q^2}, \tag{16}$$

where $f(n,m)$ is the frequency of edges linking nodes with degree $n+1$ and $m+1$, q_n is the probability of a link to connect to a node with degree $n+1$, i.e. $q_n = \sum_m f(n,m)$ and σ_q is the standard deviation of the distribution q_x.

- unweighted_mean_clustering: average clustering coefficient of the unweighted network

$$C_{uw} = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{k_i(k_i-1)} \sum_{j,h}^{1,N} L_{ij}L_{jh}L_{hi}; \tag{17}$$

- weighted_mean_clustering: average clustering coefficient of the weighted network

$$C_w = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{k_i(k_i-1)} \sum_{j,h}^{1,N} \left(W_{ij}W_{jh}W_{hi}\right)^{\frac{1}{3}}; \tag{18}$$

- network_transitivity: transitivity of the networks

$$T = \frac{2}{\sum_{i=1}^{N} k_i(k_i-1)} \sum_{i,j,h}^{1,N} L_{ij}L_{jh}L_{hi}. \tag{19}$$

The network-related features are measured via the *NetworkX* library [47].

The set of 21 features has been reduced in order to avoid redundancy due to the presence of highly correlated features. In this instance, the least amount of features such that any remaining couple has absolute Pearson correlation < 0.75 was removed. Linking the highly correlated features in an undirected unweighted network, this problem results to be equivalent to a minimum vertex cover problem (pruning the least amount of nodes such that each remaining node is isolated), therefore exactly solvable via integer linear programming (ILP). The corresponding ILP formulation, with E the set of edges,

$$\begin{aligned} &\text{Given: } x_i = \begin{cases} 1, & \text{if node } i \text{ is removed} \\ 0, & \text{otherwise} \end{cases} \\ &\text{Minimize: } \sum_{i=1}^{N} x_i \\ &\text{Subject to: } x_i + x_j \geq 1, \quad \forall (i,j) \in E \end{aligned} \tag{20}$$

was solved via the *PuLP* modeler [48]. The remaining features were 15: pca_stretch_ratio, pca_flatten_ratio, pca_alignment_score, bs_sasa_ratio, ab_bs_size, ag_bs_size, pca_stretch_ratio_bs, pca_flatten_ratio_bs, bs_mean_hydrophobicity, mean_strength, network_radius, mean_assortativity, weighted_mean_clustering, network_transitivity.

For processing, each feature was normalized via the *scikit-learn* library [49], such that all the features share the same weight. The PCA was performed via *scikit-learn*, as well.

4.4 Neural network architecture and optimization

Every NN in this work has been defined via *Tensorflow* [50] and has the same structure: two-hidden-layers feed forward NN. Each hidden layer has *reLu* activaction function, furthermore, the output layer of the NNs used in Section 2.5 are provided with a sigmoid activation function, in order to retrieve $pDockQ \in [0,1]$. While varying the number of parameters the proportion of nodes in the first and second hidden layer is kept fixed at 1:2. Therefore, naming M the number of first layer nodes, one can retrieve the number of parameters N:

$$N = (F+1)M + \frac{1}{2}M^2, \tag{21}$$

where $F = 15$ is the number of input features. The NN weights are fitted via AdamW algorithm with learning rate 0.001, through 300 epochs for the classifiers (Section 2.4) and 400 epochs for the predictor NNs (Section 2.5). In Section 2.4 binary cross-entropy was used as loss function:

$$C\left(p_{pred} \| p_{true}\right) = -\sum_{i \in \{0,1\}} p_{pred}(i)\, ln\left[p_{true}(i)\right] \tag{22}$$

where $\{0,1\}$ is the set of the possible classifications, i.e. "Decoy" or "Native-like". In Section 2.5 mean square error (MSE) was the loss function, instead:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N} \left(pDockQ_i - DockQ_i\right)^2, \tag{23}$$

where N is the number of docking predictions in the dataset and $pDockQ_i$ and $DockQ_i$ are the values of pDockQ and DockQ associated to the i-th prediction. In Section 2.5, in order to obtain the pDockQ values for the whole DINL dataset, it has been split into several complementary subsets, according to the proportion of the training set. The pDockQ values of each subset have been computed using the others as training set.

4.5 Statistical analysis

The area under the receiver operating characteristic curve (ROC AUC) was used to assess the quality of the classifications throughout Section 2.4. Given two classes (Positive and Negative) and the distribution of a measure for each of the classes, the ROC curve is the parametric curve ROC curve = $(fpr(t), tpr(t))$ representing the variation of the false positive rate $fpr(t)$ and the true positive rate $tpr(t)$ in function of the measure threshold t used to split the classes, where

$$\begin{aligned} tpr(t) &= \frac{\#\,\text{true positives}}{\#\,\text{positives}}, \\ fpr(t) &= \frac{\#\,\text{false positives}}{\#\,\text{negatives}}. \end{aligned} \tag{24}$$

The ROC AUC equates the probability that, given a random negative element and a random positive element, the negative element correspond to a measure larger than the positive. In this instance the Negative and the Positive classes were "Decoy" and "Native-like". The ROC curves and the ROC AUCs were computed via the *scikit-learn* library [49].

Regarding the regression tasks (Section 2.5), the assessment was done via Pearson correlation coefficient (ρ) between any measure x and the *DockQ* score of the docking prediction:

$$\rho(x) = \frac{\langle (x - \langle x \rangle)(DockQ - \langle DockQ \rangle) \rangle}{\sigma_x \sigma_{DockQ}}. \tag{25}$$

The validity of the Pearson correlation was assessed performing a p-value test of the null hypothesis that the distributions underlying the samples are uncorrelated and normally distributed. Both the Pearson correlation coefficient and the p-value were computed via the *SciPy* library [51].

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

AM: Investigation, Writing – original draft, Data curation, Software. GR: Writing – original draft, Conceptualization, Supervision. EM: Supervision, Conceptualization, Investigation, Writing – original draft.

Funding

The author(s) declared that financial support was received for this work and/or its publication. This research was partially funded by grants from ERC-2019-Synergy Grant (ASTRA, n. 855,923); EIC-2022-PathfinderOpen (ivBM-4PAP, n. 101098989); Project 'National Center for Gene Therapy and Drugs based on RNA Technology' (CN00000041) financed by NextGeneration EU PNRRMUR—M4C2—Action 1.4—Call 'Potenziamento strutture di ricerca e creazione di campioni nazionali di R&S' (CUP J33C22001130001); MUR PRIN 2022 (CUP: B53D2300399 0006) to EM.

Conflict of interest

The author(s) declared that this work was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The handling editor SS declared a past co-authorship with the author GR.

The authors EM, GR declared that they were an editorial board member of Frontiers at the time of submission. This had no impact on the peer review process and the final decision.

Generative AI statement

The author(s) declared that generative AI was not used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

References

1. Peace Chinedu-Nzereogu O, Atoyebi TO, Adebayo MA, Kenneth Maduike I, Alebel Dejene T, Tochukwu Excellent Okechukwu, and Yetunde Victoria. Harnessing ai-driven crispr bioinformatics: transforming precision diagnostics for antimicrobial resistance and chemical pathology. (2025).

2. Lin B, Luo X, Liu Y, Jin X. A comprehensive review and comparison of existing computational methods for protein function prediction. *Brief Bioinform* (2024) 25(4):bbae289. doi:10.1093/bib/bbae289

3. Bettanti A, Beccari AR, Biccarino M. Exploring the future of biopharmaceutical drug discovery: can advanced ai platforms overcome current challenges? *Discover Artif Intelligence* (2024) 4(1):102. doi:10.1007/s44163-024-00188-3

4. Callaway E. 'it will change everything': deepmind's ai makes gigantic leap in solving protein structures. *Nature* (2020) 588(7837):203–5. doi:10.1038/d41586-020-03348-4

5. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly accurate protein structure prediction with alphafold. *Nature* (2021) 596(7873):583–9. doi:10.1038/s41586-021-03819-2

6. Wayment-Steele HK, Ojoawo A, Otten R, Apitz JM, Pitsawong W, Hömberger M, et al. Predicting multiple conformations *via* sequence clustering and alphafold2. *Nature* (2024) 625(7996):832–9. doi:10.1038/s41586-023-06832-9

7. Baek M, DiMaio F, Anishchenko I, Dauparas J, Ovchinnikov S, Lee GR, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science* (2021) 373(6557):871–6. doi:10.1126/science.abj8754

8. Zhang Y, Luo M, Wu P, Wu S, Lee T-Y, Bai C. Application of computational biology and artificial intelligence in drug design. *Int Journal Molecular Sciences* (2022) 23(21):13568. doi:10.3390/ijms232113568

9. Norman RA, Ambrosetti F, Bonvin AMJJ, Colwell LJ, Kelm S, Kumar S, et al. Computational approaches to therapeutic antibody design: established methods and emerging trends. *Brief Bioinformatics* (2020) 21(5):1549–67. doi:10.1093/bib/bbz095

10. Ambrosetti F, Olsen TH, Paolo Olimpieri P, Jiménez-García B, Milanetti E, Marcatilli P, et al. proabc-2: prediction of antibody contacts v2 and its application to information-driven docking. *Bioinformatics* (2020) 36(20):5107–8. doi:10.1093/bioinformatics/btaa644

11. Milanetti E, Miotto M, Di Rienzo L, Monti M, Gosti G, Ruocco G. 2d zernike polynomial expansion: finding the protein-protein binding regions. *Comput Structural Biotechnology Journal* (2021) 19:29–36. doi:10.1016/j.csbj.2020.11.051

12. Di Rienzo L, Milanetti E, Lepore R, Paolo Olimpieri P, Tramontano A. Superposition-free comparison and clustering of antibody binding sites: implications for the prediction of the nature of their antigen. *Scientific Reports* (2017) 7(1):45053. doi:10.1038/srep45053

13. Di Rienzo L, Miotto M, Desantis F, Grassmann G, Ruocco G, Milanetti E. Dynamical changes of sars-cov-2 spike variants in the highly immunogenic regions impact the viral antibodies escaping. *Proteins: Struct Funct Bioinformatics* (2023) 91(8):1116–29. doi:10.1002/prot.26497

14. Abramson J, Adler J, Dunger J, Evans R, Green T, Pritzel A, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature* (2024) 630(8016):493–500. doi:10.1038/s41586-024-07487-w

15. Ma Nooren I, Thornton JM. Diversity of protein–protein interactions. *The EMBO Journal* (2003). doi:10.1093/emboj/cdg359

16. Grassmann G, Miotto M, Desantis F, Di Rienzo L, Tartaglia GG, Pastore A, et al. Computational approaches to predict protein–protein interactions in crowded cellular environments. *Chem Rev* (2024) 124(7):3932–77. doi:10.1021/acs.chemrev.3c00550

17. Zhang K, Tao Y, Wang F. Antibinder: utilizing bidirectional attention and hybrid encoding for precise antibody–antigen interaction prediction. *Brief Bioinform* (2024) 26(1):bbaf008. doi:10.1093/bib/bbaf008

18. De Lauro A, Di Rienzo L, Miotto M, Paolo Olimpieri P, Milanetti E, Ruocco G. Shape complementarity optimization of antibody–antigen interfaces: the application to sars-cov-2 spike protein. *Front Mol Biosciences* (2022) 9:874296. doi:10.3389/fmolb.2022.874296

19. Li M, Shi Y, Hu S, Hu S, Guo P, Wan W, et al. Mvsf-ab: accurate antibody–antigen binding affinity prediction *via* multi-view sequence feature learning. *Bioinformatics* (2025) 41(5):btae579. doi:10.1093/bioinformatics/btae579

20. Michalewicz K, Barahona M, Bravi B. Antipasti: interpretable prediction of antibody binding affinity exploiting normal modes and deep learning. *Structure* (2024) 32(12):2422–34. doi:10.1016/j.str.2024.10.001

21. Kozakov D, Hall DR, Xia B, Porter KA, Padhorny D, Yueh C, et al. The cluspro web server for protein–protein docking. *Nat Protocols* (2017) 12(2):255–78. doi:10.1038/nprot.2016.169

22. Jiménez-García B, Roel-Touris J, Barradas-Bautista D. The lightdock server: artificial intelligence-powered modeling of macromolecular interactions. *Nucleic Acids Research* (2023) 51(W1):W298–W304. doi:10.1093/nar/gkad327

23. Chen R, Li L, Weng Z. Zdock: an initial-stage protein-docking algorithm. *Proteins: Struct Funct Bioinformatics* (2003) 52(1):80–7. doi:10.1002/prot.10389

24. Yan Y, Tao H, He J, Huang S-Y. The hdock server for integrated protein–protein docking. *Nat Protocols* (2020) 15(5):1829–52. doi:10.1038/s41596-020-0312-x

25. Dominguez C, Boelens R, Bonvin AMJJ. Haddock: a protein-protein docking approach based on biochemical or biophysical information. *J Am Chem Soc* (2003) 125(7):1731–7. doi:10.1021/ja026939x

26. Zhao N, Han B, Zhao C, Xu J, Gong X. Abag-docking benchmark: a non-redundant structure benchmark dataset for antibody–antigen computational docking. *Brief Bioinform* (2024) 25(2):bbae048. doi:10.1093/bib/bbae048

27. Ambrosetti F, Jiménez-García B, Roel-Touris J, Bonvin AMJJ. Modeling antibody-antigen complexes by information-driven docking. *Structure* (2020) 28(1):119–29. doi:10.1016/j.str.2019.10.011

28. Vittorio S, Lunghini F, Morerio P, Gadioli D, Orlandini S, Silva P, et al. Addressing docking pose selection with structure-based deep learning: recent advances, challenges and opportunities. *Comput Struct Biotechnol J* (2024) 23:2141–51. doi:10.1016/j.csbj.2024.05.024

29. Dong L, Liu J, Wang H, Liang F, Zhang G. Deepumqa-x: comprehensive and insightful estimation of model accuracy for protein single-chain and complex. *Nucleic Acids Res* (2025) 53(W1):W219–W227. doi:10.1093/nar/gkaf380

30. Basu S, Wallner B. Dockq: a quality measure for protein-protein docking models. *PLOS ONE* (2016) 11(8):e0161879. doi:10.1371/journal.pone.0161879

31. Lensink MF, Wodak SJ. Docking, scoring, and affinity prediction in capri. *Proteins* (2013) 81(12):2082–95. doi:10.1002/prot.24428

32. Collins KW, Copeland MM, Brysbaert G, Wodak SJ, Bonvin AMJJ, Kundrotas PJ, et al. Capri-q: the capri resource evaluating the quality of predicted structures of protein complexes. *J Molecular Biology* (2024) 436(17):168540. doi:10.1016/j.jmb.2024.168540

33. Graber D, Stockinger P, Meyer F, Mishra S, Horn C, Buller R. Resolving data bias improves generalization in binding affinity prediction. *Nat Machine Intelligence* (2025) 7:1713–25. doi:10.1038/s42256-025-01124-5

34. Pellicani F, Ben DD, Perali A, Pilati S. Machine learning scoring functions for drug discovery from experimental and computer-generated protein–ligand structures: towards per-target scoring functions. *Molecules* (2023) 28(4):1661. doi:10.3390/molecules28041661

35. Yang J, Shen C, Huang N. Predicting or pretending: artificial intelligence for protein-ligand interactions lack of sufficiently large and unbiased datasets. *Front Pharmacol* (2020) 11–2020. doi:10.3389/fphar.2020.00069

36. Ahmad I, Jadhav H, Shinde Y, Jagtap V, Girase R, Patel H. Optimizing bedaquiline for cardiotoxicity by structure based virtual screening, dft analysis and molecular dynamic simulation studies to identify selective mdr-tb inhibitors. *Silico Pharmacol* (2021) 9(1):23. doi:10.1007/s40203-021-00086-x

37. Alandijany TA, El-Daly MM, Ahmed MT, Bajrai LH, Khateb AM, Alsaady IM, et al. Investigating the mechanism of action of anti-dengue compounds as potential binders of zika virus rna-dependent rna polymerase. *Viruses* (2023) 15(7):1501. doi:10.3390/v15071501

38. Huang SY, Zou X. An iterative knowledge-based scoring function for protein-protein recognition. *Proteins* (2008) 72(2):557–79. doi:10.1002/prot.21949

39. Hou T, Wang J, Li Y, Wang W. Assessing the performance of the Mm/pbsa and mm/gbsa methods. 1. the accuracy of binding free energy calculations based on molecular dynamics simulations. *J Chem Inf Model* (2011) 51(1):69–82. doi:10.1021/ci100275a

40. Zhang X, Jiang L, Weng G, Shen C, Zhang O, Liu M, et al. Hawkdock version 2: an updated web server to predict and analyze the structures of protein–protein complexes. *Nucleic Acids Res* (2025) 53(W1):W306–W315. doi:10.1093/nar/gkaf379

41. Dunbar J, Krawczyk K, Leem J, Baker T, Fuchs A, Georges G, et al. Sabdab: the structural antibody database. *Nucleic Acids Research* (2014) 42(D1):D1140–D1146. doi:10.1093/nar/gkt1043

42. Fu L, Niu B, Zhu Z, Wu S, Li W. Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics* (2012) 28(23):3150–2. doi:10.1093/bioinformatics/bts565

43. Li W, Godzik A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* (2006) 22(13):1658–9. doi:10.1093/bioinformatics/btl158

44. Huang Y, Niu B, Gao Y, Fu L, Li W. Cd-hit suite: a web server for clustering and comparing biological sequences. *Bioinformatics* (2010) 26(5):680–2. doi:10.1093/bioinformatics/btq003

45. Bonella S, Raimondo D, Milanetti E, Tramontano A, Ciccotti G. Mapping the hydropathy of amino acids based on their local solvation structure. *The J Phys Chem B* (2014) 118(24):6604–13. doi:10.1021/jp500980x

46. Cock PJA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics* (2009) 25(11):1422–3. doi:10.1093/bioinformatics/btp163

47. Hagberg AA, Schult DA, Swart PJ. Exploring network structure, dynamics, and function using networkx. In: G Varoquaux, T Vaught, J Millman, editors. *Proceedings of the 7th python in science conference*. Pasadena, CA USA (2008). p. 11–5.

48. Mitchell S, O'Sullivan M, Dunning I. Pulp: a linear programming toolkit for python. (2011).

49. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: machine learning in Python. *J Machine Learn Res* (2011) 12:2825–30.

50. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. Tensorflow: a system for large-scale machine learning. In: *Proceedings of the 12th USENIX conference on operating systems design and implementation, OSDI'16*. USA: USENIX Association (2016). p. 265–83.

51. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: fundamental algorithms for scientific computing in python. *Nat Methods* (2020) 17:261–72. doi:10.1038/s41592-019-0686-2