BMC Bioinformatics

RESEARCH Open Access

Struct2Graph: a graph attention network for structure based predictions of protein–protein interactions

Mayank Baranwal[1,2*], Abram Magner[3], Jacob Saldinger[4], Emine S. Turali-Emre[5†], Paolo Elvati[6†], Shivani Kozarekar[4], J. Scott VanEpps[5,7,8], Nicholas A. Kotov[4,5,8,9], Angela Violi[4,6,10] and Alfred O. Hero[5,11,12,13,14]

†Emine S. Turali-Emre and Paolo Elvati equal contributor

*Correspondence: baranwal.mayank@tcs.com

[2] Systems and Control Engineering Group, Indian Institute of Technology, Bombay, India
Full list of author information is available at the end of the article

Abstract

Background: Development of new methods for analysis of protein–protein interactions (PPIs) at molecular and nanometer scales gives insights into intracellular signaling pathways and will improve understanding of protein functions, as well as other nanoscale structures of biological and abiological origins. Recent advances in computational tools, particularly the ones involving modern deep learning algorithms, have been shown to complement experimental approaches for describing and rationalizing PPIs. However, most of the existing works on PPI predictions use protein-sequence information, and thus have difficulties in accounting for the three-dimensional organization of the protein chains.

Results: In this study, we address this problem and describe a PPI analysis based on a graph attention network, named *Struct2Graph*, for identifying PPIs directly from the structural data of folded protein globules. Our method is capable of predicting the PPI with an accuracy of 98.89% on the balanced set consisting of an equal number of positive and negative pairs. On the unbalanced set with the ratio of 1:10 between positive and negative pairs, Struct2Graph achieves a fivefold cross validation average accuracy of 99.42%. Moreover, Struct2Graph can potentially identify residues that likely contribute to the formation of the protein–protein complex. The identification of important residues is tested for two different interaction types: (a) Proteins with multiple ligands competing for the same binding area, (b) Dynamic protein–protein adhesion interaction. Struct2Graph identifies interacting residues with 30% sensitivity, 89% specificity, and 87% accuracy.

Conclusions: In this manuscript, we address the problem of prediction of PPIs using a first of its kind, 3D-structure-based graph attention network (code available at https://github.com/baranwa2/Struct2Graph). Furthermore, the novel mutual attention mechanism provides insights into likely interaction sites through its unsupervised knowledge selection process. This study demonstrates that a relatively low-dimensional feature embedding learned from graph structures of individual proteins outperforms other modern machine learning classifiers based on global protein features. In addition, through the analysis of single amino acid variations, the attention mechanism shows preference for disease-causing residue variations over benign polymorphisms, demonstrating that it is not limited to interface residues.

Keywords: Protein–protein interaction, Deep learning, Structure-based prediction, Graph attention network

Introduction

Protein–protein interactions (PPIs) are fundamental to many biological processes. Analysis of the human proteome suggests that the majority of proteins function not alone but rather as part of multi-unit complexes [1]. Indeed, PPIs are the central part of signal transduction, metabolic regulation, environmental sensing, and cellular organization [2]. In these processes, PPIs can alter enzyme kinetics, facilitate substrate channeling, form new binding sites, render a protein inactive, or modify the specificity of a protein with respect to a substrate [3]. Due to the ubiquitous presence of PPIs in living systems, being able to characterize these interactions promises to further our understanding of cellular processes [4] and provide an indispensable tool for disease treatment and drug discovery [5, 6]. PPI and their mathematical description are also essential for creation of protein analogs from other nanoscale building blocks, including but not limited to, lipids [7], sugars [8], polymers [9], nanoscale conjugates [10], and inorganic nanoparticles [11–13].

A number of strategies have been employed to decode PPIs aiming primarily at molecular scale data and amino acid sequences [14]. Traditionally, high throughput experimental techniques such as two-hybrid screens [15], tandem-affinity purification [16], and mass spectrometry [17] have been applied to create protein interaction networks. Concerns about insufficient accuracy [18], low experimental throughput [19] and high cost [20] of these methods, however, have motivated computational approaches that can complement traditional and robotic experimental protocols. Computational methods can predict whether proteins will interact based on data for the proteins' genetic context, amino acid sequences, or structural information. Genomics analyses consider factors such as gene fusion [21], conservation across common species (phylogenetic profiling) [22], and evolutionary history [23] when determining if a pair of proteins interact.

Typical computational techniques for PPI analysis use the amino acid sequences of the two proteins to determine whether interactions occur [24, 25]. A number of features such as frequency of common sub-sequences [26] and auto-covariance [27] have been proposed to convert sequences of different lengths into a uniformly sized representation. Sequence based methods have recently been able to leverage protein databases and machine learning techniques to make high accuracy predictions. Three-dimensional (3D) structure of protein–protein complexes from sequence can be predicted by CO-threading algorithm, (COTH) that recognizing templates of protein complexes from solved complex structural databases. COTH aligns amino acid chain sequences using scoring functions and structural information [28]. The DeepPPI model [29] predicts interactions using an artificial neural network, which takes as input a feature vector that captures the composition, distribution, and order of the sequence. DeepFE [30] uses natural language processing algorithms on amino acid sequences to create low dimensional embeddings of the sequence suitable as inputs for neural network analysis. DeepFE, in particular, has been shown to be quite effective, and achieves prediction accuracy of 94.78% and 98.77% on *S. cerevisiae* and human datasets, respectively. In fact, most deep learning based methods have been shown to achieve high PPI prediction accuracy [31, 32] owing to their significantly larger representation power. In addition to relying purely on sequence-based information, modern machine learning methods often incorporate network-level information for PPI prediction. In a PPI network, each node represents a protein, while edges between them represent interactions. Thus, predicting interactions between any two nodes is a link-prediction problem in disguise. Recent methods have leveraged the network structure, along with using vectorized representation of amino acid sequences, to obtain stronger prediction performance [13, 33–37].

Despite their success, the above sequence based approaches do not generalize to broader classes of chemical compounds of similar scale as proteins that are equally capable of forming complexes with proteins that are not based on amino acids, and thus lack of an equivalent sequence-based representation. While the interaction of proteins with DNA can be accurately predicted [38], the methods for machine learning-based predictions for protein complexes with high molecular weight lipids [7], sugars [8], polymers [9], dendrimers [39] and inorganic nanoparticles [11, 12] that receive much attention in nanomedicine and nanodiagnostics [40, 41], are not widely known among experimentalists [42–48], although substantial strides in this direction were made with the development of unified structural descriptors for proteins and nanoparticles [13]. As a consequence, predictive computational approaches that take into account the structure of proteins and their variable non-proteinatious, biomimetic, and non-biological counterparts become possible. Some methods predict interactions using the 3D structure of the proteins [49, 50] use a knowledge-based approach to assess the structural similarity of candidate proteins to a template protein complex. As this methodology requires detailed information on the larger complex, template-free docking approaches [51] analyze the unbound protein components and identify the most promising interactions from a large set of potential interaction sites. While docking methods have shown success for some proteins, they face difficulty with proteins undergoing conformational changes during interaction [52]. Many of these structural approaches have also served as the basis for machine learning models. Zhang et al. developed PrePPI [53] which uses amino acid sequence, and phylogenetic features as inputs to a naive Bayes classifier. Northey et al. developed IntPred [54] which segments proteins into a group of patches that incorporates 3D structural information into a feature set to predict interaction with a multi-layer perception network. These models are trained on carefully curated interaction databases describing both binary interactions between proteins, and corresponding interfacing sites or atoms.

In this work, we make the first step toward a generalized method to assess supramolecular interactions of proteins with other nanostructures. The proposed method determines the probability of formation of protein–protein complexes on a nanoscale representation of proteins from crystallographic data, as contrasted to amino-acid amino sequence information. We develop a mutual graph attention network and a corresponding computational tool, *Struct2Graph*, to predict PPIs solely from 3D structural information. Instead of using several protein specific features, such as, hydrophobicity, solvent accessible surface area (SASA), charge, frequency of *ngrams*, etc., Struct2Graph uses a graph based representation of a protein globule obtained using *only* the 3D positions of atoms. This graph based interpretation allows for neural message passing [55] for efficient representation learning of proteins. Struct2Graph builds upon our prior work on metabolic pathway prediction [56], where it is shown that an equivalent graph-based structural representation of small molecules and peptides coupled with graph convolutional network, significantly outperforms other classifiers that involve computing various biochemical features as inputs. This approach also leverages generalization of graph theory to describe complex nanoscale assemblies similar to PPI [57].

Beyond the high accuracy of its PPI predictions, Struct2Graph offers a number of advantages. Similarly to the ML algorithms exploiting the idea of geometrical biomimetics, Struct2Graph only requires the 3D structure of individual proteins. Furthermore, while in this paper we focus on protein interactions, by using only the positions of atoms in our analysis, this framework can be generalized to other molecular structures where 3D information is available. Moreover, Struct2Graph is also able to provide insight into the nature of the protein interactions. Through its attention mechanism, the model can potentially identify residues that likely contribute to the formation of the protein–protein complex. Unlike other models, Struct2Graph is able to produce this data in an unsupervised manner and thus does not require protein complex information which are often unavailable [58].

The key contributions of the proposed work can be summarized as:

- *Graph convolutional network for PPI prediction*: Struct2Graph uses a multi-layer graph convolutional network (GCN) for PPI prediction from the structural data of folded protein globules. The proposed approach is general and can be applied to other nanoscale structures where 3D information is available.
- *Curation of PPI database*: A large PPI database comprising of only direct/physical interaction of *non-homologous* protein pairs[1] is curated, along with information on the corresponding PDB files. Special emphasis is based on curation of PDB files based on the length of the chain ID and highest resolution within each PDB file to ensure capturing of the most complete structure information of the protein of interest.
- *State-of-the-art prediction performance*: Our method is capable of correctly predicting the PPIs with an accuracy of 98.89% on the balanced set consisting of an equal number of positive and negative pairs. On the unbalanced set with the ratio of 1:10 between positive and negative pairs, Struct2Graph achieves a fivefold cross validation average accuracy of 99.42%. Struct2Graph outperforms not only the classical feature-based machine learning approaches, but also other modern deep-learning approaches, such as Deep-PPI and DeepFE-PPI that use sequence information and feature selection for PPI prediction.
- *Unsupervised prediction of important residues*: The novel mutual attention mechanism can potentially identify important residues for the formation of the protein–protein complex. This importance can stem from either direct participation in the interaction process (i.e., binding site) or indirectly through contribution to appropri-

[1] Based on the pairwise homology analysis comprising of 3677 unique proteins in our database, only 0.3% of the proteins were found to have BLAST e-value < 0.05 and 0.26% has < 0.001, indicating statistically insignificant homologous relationships.

ate protein folding that allows formation of the correct binding site geometry. The identification of important residues is tested for two different interaction types (neither part of the training set): (a) Proteins with multiple ligands competing for the same binding area, (b) Dynamic protein–protein adhesion interaction. Struct2Graph identifies interacting residues with 30% sensitivity, 89% specificity, and 87% accuracy.

- *Analysis of single amino acid variation (SAV) dataset*: Disease-causing mutations are known to be preferentially located within the interface core, as opposed to the rim. Of the known 2724 disease-causing SAVs and 1364 polymorphisms, our attention mechanism identifies 33.55% of all disease-causing SAVs as important (attention weights within top-20%), while 85.30% of all polymorphisms are identified as *un*important by the proposed attention mechanism, indicating significant overlap between the previously established SAV study and the important residues identified by the proposed attention mechanism.

Materials and methods

PPI database

Struct2Graph focuses on structure-based predictions and interaction sites of the protein pairs. Our PPI database is therefore produced based on only direct/physical interactions of proteins excluding therefore weakly interacting and loosely associated nanoscale biomolecules. To build a large physical interaction database, comprising of only *heterologous* pairs, we searched all possible databases available (STRING, BioGRID, IntAct, MINT, BIND, DIP, HPRD, APID, OpenWetWare). Not all PPI databases use the same publications and same ontologies to report the interactions. Consequently, it is not surprising that each database reports PPI differently. Therefore, only up to a 75% concordance between all PPI databases is achieved [59]. For Struct2Graph, two of the largest compiled databases, IntAct [60] and STRING [61] are chosen for further analysis, and results are compared to each other to find the true interactions. Only concordant matches between these two databases are chosen. Struct2Graph database is compiled from commonly studied organism's (*Saccharomyces cerevisiae*, *Homo sapiens*, *Escherichia coli*, *Caenorhabditis elegans* and *Staphylococcus aureus*) PPIs. For these organisms, IntAct provides 427,503 PPIs, and STRING provides 852,327 PPIs.

STRING distinguishes the type of interactions as "activation", "binding", "catalysis", "expression", "inhibition" and "reaction". IntAct, on the other hand, describe the type of interactions as "association", "physical association", "direct association/interaction", and "colocalization". Only "direct association/interactions" from IntAct and "binding" from STRING were considered as physical interactions. We only choose concordant pairs of physical interactions from both databases. Therefore, extracting only physical interaction data from the rest of the interactions reduces the actual number of PPIs to 12,676 pairs for IntAct and 446,548 pairs for STRING. Negative PPI is extracted from the work that derives negative interaction from large-scale two-hybrid experiments [62]. The negative protein–protein pairs from the two-hybrid system are compared further with the database constructed from STRING and INTact, and only the pairs that are not involved in any interaction at all, are chosen. We further exclude the co-localized protein pairs in our analysis. Structure information for Struct2Graph is obtained from PDB files. Hence,

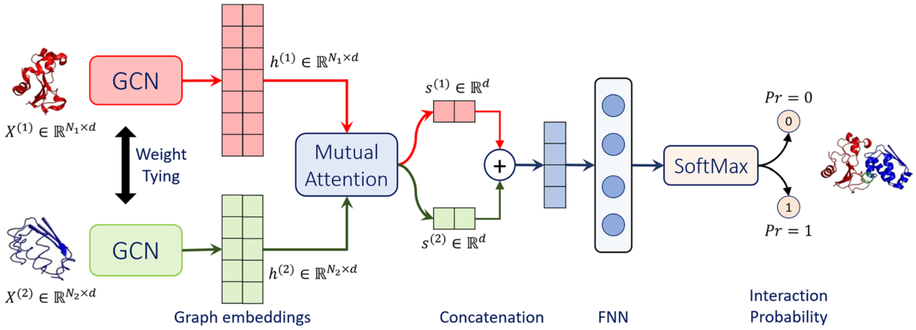
GCN
$h^{(1)} \in \mathbb{R}^{N_1 \times d}$
$X^{(1)} \in \mathbb{R}^{N_1 \times d}$
Weight Tying
GCN
$h^{(2)} \in \mathbb{R}^{N_2 \times d}$
$X^{(2)} \in \mathbb{R}^{N_2 \times d}$
Mutual Attention
$s^{(1)} \in \mathbb{R}^{d}$
$s^{(2)} \in \mathbb{R}^{d}$
+
SoftMax
$Pr = 0$
0
1
$Pr = 1$
Graph embeddings
Concatenation
FNN
Interaction Probability

Fig. 1 Struct2Graph schematic. Struct2Graph graph convolutional network (GCN) for incorporating mutual attention for PPI prediction. The GCN classifies whether or not a protein pair ($X^{(1)}$ and $X^{(2)}$ on far left) interacts and predicts the interaction sites (on far right)

we only used the pairs which have associated PDB files. This reduces the total number of pairs to 117,933 pairs (4698 positive and 112,353 negatives). Some proteins are well-studied as they are in the scope of current medical and biotechnological interest. As a result, there is more than one cross-reference to PDB files since various structures are accessible for these proteins. To find the proteins matched with PDB files, all proteins from the database are matched with UniProt accession numbers (UniProt Acc) and mapped with PDB files in UniProt [63]. Unfortunately, not all proteins are crystallized fully in each PDB file, and random choice of PDB file may cause incomplete information of the binding site of the protein. Therefore, we curated the PDB files based on the length of the chain ID and highest resolution within each PDB file to ensure that we capture the most complete structure information of the protein of interest. The chain length and the resolution of each protein's crystal structure were obtained from the RCSB website [64]. The complete set of negative pairs was reduced to 5036 pairs to create a fairly balanced training sample with an approximately equal number of positive and negative pairs. For this curated database consisting of only heterologous pairs, we defined two classes, "0" for non-interacting (negative: not forming complexes) pairs and "1" for interacting (positive: forming complexes) pairs.

Mutual graph attention network for protein–protein pairs

We present a novel multi-layer mutual graph attention network (GAT) based architecture for PPI prediction task, summarized in Fig. 1. We refer to this architecture as *Struct2Graph*, since the inputs to the proposed GAT are coarse grained structural descriptors of a query protein–protein pair. Struct2Graph outputs the probability of interaction between the query proteins. Struct2Graph uses two graph convolutional networks (GCNs) with weight sharing, and a mutual attention network to extract relevant geometric features related to query protein pairs. These extracted features are then concatenated and fed to a feedforward neural network (FNN) coupled with a `SoftMax` function, which finally outputs a probability of the two classes—'0' (negative pairs) and '1' (positive pairs). This section first describes the preprocessing and fingerprinting procedure specifying how spatial information on protein pairs are converted into corresponding protein graphs, and then elaborates on different components of the Struct2Graph deep learning architecture.

Protein structure graph

The purpose of the graph construction step is to capture the salient geometry of the proteins in a way that is amenable to further dimensionality reduction by the neural network. There are many possible ways of constructing a graph from spatial coordinates of individual atoms, and each captures a different level of detail about the geometry of the protein [13]. For instance, the protein-contact graph described in [65] adds edges between three nearest neighbors and identifies the nodes as helices, sheets, and turns. Ralaivola et al. [66] uses molecular fingerprints to prescribe contact graphs for chemical compounds. Pires et al. [67] employs a distance-based approach for constructing protein graphs by encoding distance patterns between constituent atoms. Cha et al. [13] pioneered multidimentional protein graphs with embedded chemical, geometrical and graph theoretical descriptors. Our approach to constructing protein graphs is inspired by the latter, however, it is generalizable to other non-protein structures as well. We first aggregate atoms into the amino acids that they constitute and define the position of an amino acid to be the average of the positions of its constituent atoms. These amino acids form the vertices of the protein graph. An edge is placed between two vertices if the distance between them is less than some threshold. Unlike the previous studies with 7Å threshold [13], in this work, we use a threshold of 9.5Å for creating a protein graph from the mean positions of amino acids. This threshold was obtained empirically so as to render the underlying graph fully connected, while simplifying the graph representation. Note that while we use amino acids as constituent vertices of the protein graphs, the approach can be easily extended to multiresolution representation, where a vertex represents two or more amino acids. The coarse-grained representation opens up new possibilities for studying other nanoscale components of protein complexes, such as, lipids and polysaccharides, since, lowering the level of representation from all-atom to submolecular can be easily generalized to other non-protein entities. Graphs with greater structural refinement can also be obtained by using functional groups as amino acids. Moreover, this geometric construction of protein graphs ensures that salient geometric features, such as spatial proximity of non-adjacent amino acids along the polypeptide chain are captured. A sequence based representation of proteins might not capture this geometrical structure as well (see Fig. 2).

The graph construction approach converts spatial information associated with a protein into an equivalent protein graph object $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of edges between them. In the context of protein graph in Fig. 2, $v_i \in \mathcal{V}$ is the ith amino acid and $e_{ij} \in \mathcal{E}$ represents an edge between ith and jth amino acids, satisfying their proximity within the specified threshold of 9.5Å. These graph objects must be embedded into real-valued vector space in order to employ our machine learning framework. We use 1-neighborhood subgraphs [56] induced by the neighboring vertices and edges at 1-hop distance from a vertex. A dictionary of all unique subgraphs is constructed by scanning all protein graphs in the training database. Thus, each vertex within a protein is equivalently represented by an element in the dictionary.

Graph convolutional network acting on protein graphs

A graph convolutional network (GCN) maps graphs to real-valued *embedding vectors* in such a way that the geometry of the embedding vectors reflects similarities between the

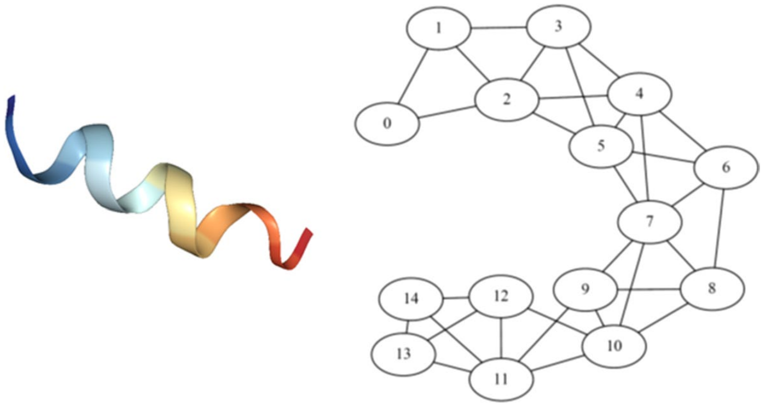

Fig. 2 Protein and protein graph. Illustration of extracted protein structure graph (right) from the corresponding PDB description of a peptide segment (left) of the *S. cerevisiae* alpha-factor receptor. The graph is extracted by thresholding the distances between amino acids. The helical structure of the protein (left) gets captured in the corresponding protein graph (right) where, for example, amino acid 4 is linked with amino acid 7

graphs. The embedding portion of the GCN works as follows. To each vertex $v_i \in \mathcal{V}$, we associate a d-dimensional feature vector, which encodes the 1-neighborhood subgraph induced by the neighboring vertices and edges at 1-hop distance from a vertex. This is in contrast to explicit inclusion of amino acid specific features, such as, hydrophobicity, solvent accessible surface area (SASA), charge, etc. In our encoding, similar to other studies [56, 68], each element of the dictionary of subgraphs is assigned a random unit-norm vector.

Each layer of the GCN updates all vertex features by first replacing each vertex feature by a normalized average over vertex features of all 1-hop neighboring vertices. This is followed by an affine transformation given by the trained weight matrices and bias parameters. In order to impart expressivity to the GCN architecture, each coordinate of the resulting affine transformed embedding vector is passed through a nonlinear activation function, such as, rectified linear unit (ReLU) or sigmoid activations. This process is repeated for all the subsequent layers, and the output of the final layer is the newly transformed embedding (feature) vector that is propagated further to the mutual attention network. Here, the number of layers is a hyperparameter, while the weight matrices are learned from the training data in order to optimize performance of the entire system on the interaction prediction task.

More concisely, given input protein graphs $\mathcal{G}^{(1)}, \mathcal{G}^{(2)}$ with adjacency matrices $A^{(1)}, A^{(2)}$ consisting of N_1, N_2 vertices (amino acids), and quantities $X_0^{(1)} \in \mathbb{R}^{N_1 \times d}$, $X_0^{(2)} \in \mathbb{R}^{N_2 \times d}$ representing the d-dimensional embedding of the vertex subgraphs of the query protein–protein pair, respectively, an l-layer GCN updates vertex embeddings using the following update rule:

$$X_{t+1}^{(m)} = \mathtt{ReLU}\left(\tilde{A}^{(m)} X_t^{(m)} W_t\right), \quad \text{for all} \quad t \in \{0, \ldots, l-1\}, \tag{1}$$

where $\tilde{A}^{(m)} = \left(\hat{D}^{(m)}\right)^{-\frac{1}{2}} \hat{A}^{(m)} \left(\hat{D}^{(m)}\right)^{-\frac{1}{2}}$ denotes the normalized adjacency matrices, and $m \in \{1, 2\}$. Here, $\hat{A}^{(m)} = A^{(m)} + I$ and $\hat{D}^{(m)}$ is the degree matrix of $\hat{A}^{(m)}$. Parameters W_t denote the weight matrix associated with the tth-layer of the GCN. The feature embeddings $X_l^{(1)} \in \mathbb{R}^{N_1 \times d}$ and $X_l^{(2)} \in \mathbb{R}^{N_2 \times d}$ produced by the final layer of GCN are fed to a mutual attention network and hereafter denoted as $h^{(1)}$ and $h^{(2)}$, respectively, for notational convenience.

Mutual attention network for PPI prediction

The purpose of the proposed mutual attention network is two fold: (a) extract relevant features for the query protein–protein pair that *mutually* contribute towards prediction of physical interaction of proteins, (b) combine embedding matrices of dissimilar dimensions $N_1 \times d$ and $N_2 \times d$ to produce a representative single output embedding vector of dimension ($2d$). Attention mechanisms were originally introduced for interpreting sequence-to-sequence translation models by allowing the models to attend differently to different parts of the encoded inputs. Since then, it has been adapted in other fields of deep learning, such as, computer vision [69], and bioinformatics [68].

The mutual attention mechanism proposed in this work computes attention weights $\left[\alpha_{ij}\right] \in \mathbb{R}^{N_1 \times N_2}$ and context vectors $s^{(1)} \in \mathbb{R}^d$, $s^{(2)} \in \mathbb{R}^d$ from the GCN-transformed hidden embeddings $h^{(1)}$ and $h^{(2)}$ (as shown in Fig. 1). The sizes of these GCN-embeddings are $N_1 \times d$ and $N_2 \times d$, respectively. For each residue i in the first protein, the GCN-embedding is denoted by $h_i^{(1)}$, which is d-dimensional. Similarly, the embedding of the jth residue in the second protein is denoted by $h_j^{(2)}$. Depending on the sizes of the input proteins, N_1 and N_2 can be arbitrary, and we want our Struct2Graph model to be invariant to the sizes N_1 and N_2. This is achieved using learnable weights U and V of sizes $d \times d$ each, and a weight vector $w \in \mathbb{R}^d$. In particular, the attention weights are computed as:

$$\alpha_{ij} = w^\top \tanh\left(Uh_i^{(1)} + Vh_j^{(2)}\right). \tag{2}$$

Here U, V and w are trained in an end-to-end fashion along with the weights of the GCN. These attention weights are then translated to context vectors $s^{(1)}, s^{(2)}$ (see Fig. 1) using the following knowledge selection procedure:

$$\begin{aligned}
\eta_i^{(1)} &= \frac{1}{N_2}\sum_{j=1}^{N_2} \alpha_{ij}, & \eta_j^{(2)} &= \frac{1}{N_1}\sum_{i=1}^{N_1} \alpha_{ij} \\
p_i^{(1)} &= \frac{\exp\left(\eta_i^{(1)}\right)}{\sum_{k=1}^{N_1} \exp\left(\eta_k^{(1)}\right)}, & p_j^{(2)} &= \frac{\exp\left(\eta_j^{(2)}\right)}{\sum_{k=1}^{N_2} \exp\left(\eta_k^{(2)}\right)}. \\
s^{(1)} &= \sum_{i=1}^{N_1} p_i^{(1)} h_i^{(1)}, & s^{(2)} &= \sum_{j=1}^{N_2} p_j^{(2)} h_j^{(2)}
\end{aligned} \tag{3}$$

From the perspective of the first protein with N_1 residues, the proposed knowledge selection process in (3) takes the column average of the matrix of attention weights, resulting into an N_1-sized vector $\eta^{(1)}$. We then perform a `SoftMax` operation that outputs probability vector $p^{(1)}$ from the intermediate embedding $\eta^{(1)}$. Finally, a d-dimensional embedding, $s^{(1)}$, is created as the weighted mean of the GCN-embedding of the first protein using probabilities of residues as the weights. A similar process is repeated for the second protein to obtain another d-dimensional embedding. Thus, while the final embedding that forms the input to the feed-forward network (FFN) is a concatenation of the context vectors $s^{(1)}$ and $s^{(2)}$, the corresponding probability vectors $p^{(1)}$ and $p^{(2)}$ capture the relative significance of the individual residues in both the proteins, respectively. Those vertices whose learned attention weights are large are likely to represent residues that participate directly or indirectly towards forming a protein–protein complex.

The context vectors $s^{(1)}$ and $s^{(2)}$ are then concatenated into a single context vector of dimensions $2d$, which is used as input to a single-layer, fully connected feedforward neural network (FNN) represented by $f(\cdot)$ to produce a two-dimensional output vector. The FNN is parameterized by another weight matrix to be learned in an end-to-end manner. A final SoftMax layer is applied to produce a probability, one for each of the possible classes: 0 or 1, as shown in Eq. (4). This output represents the classifier's prediction of the probability that the two proteins interact.

$$y_{\text{out}} = \texttt{SoftMax}\Big(f\Big(\texttt{concat}\Big[s^{(1)}, s^{(2)}\Big]\Big)\Big) \tag{4}$$

The pseudocode below (Model details) summarizes the details of the proposed Struct2Graph model.

Model details Struct2Graph

```
**Inputs**: # of GCN layers $l$, Embedding dimension $d$, Protein graphs $(X_0^{(1)}, A^{(1)}), (X_0^{(2)}, A^{(2)})$
**Model parameters**: $\{W_t\}$ for $t \in 0, \cdots, l-1$, $w$,$U$, $V$, Weights of FFN
Normalize adjacency matrices to obtain $\tilde{A}^{(1)} \in \mathbb{R}^{N_1 \times N_1}$, $\tilde{A}^{(2)} \in \mathbb{R}^{N_2 \times N_2}$ $(N_i$: #vertices of graph $i)$

# Produce GCN-embeddings
for $t = 0,\ t \leq l-1$ do
    for $m = 1,\ m \leq 2$ do
        $X_{t+1}^{(m)} \leftarrow \texttt{ReLU}\left(\tilde{A}^{(m)} X_t^{(m)} W_t\right)$
    end for
end for
$h^{(1)} \leftarrow X_l^{(1)}$, $h^{(2)} \leftarrow X_l^{(2)}$

# Mutual attention mechanism
for $i = 1,\ i \leq N_1$ do
    for $j = 1,\ j \leq N_2$ do
        $\alpha_{ij} \leftarrow w^{\intercal} \texttt{tanh}\left(U h_i^{(1)} + V h_j^{(2)}\right)$
    end for
end for

# Knowledge selection process
for $i = 1,\ i \leq N_1$ do
    $\eta_i^{(1)} \leftarrow \frac{1}{N_2} \sum_{j=1}^{N_2} \alpha_{ij}$;        $p_i^{(1)} \leftarrow \frac{\exp \eta_i^{(1)}}{\sum_{k=1}^{N_1} \exp \eta_k^{(1)}}$
end for
$s^{(1)} \leftarrow \sum_{i=1}^{N_1} p_i^{(1)} h_i^{(1)}$
for $j = 1,\ j \leq N_2$ do
    $\eta_j^{(2)} \leftarrow \frac{1}{N_1} \sum_{i=1}^{N_1} \alpha_{ij}$;        $p_j^{(2)} \leftarrow \frac{\exp \eta_j^{(2)}}{\sum_{k=1}^{N_2} \exp \eta_k^{(2)}}$
end for
$s^{(2)} \leftarrow \sum_{j=1}^{N_2} p_j^{(2)} h_j^{(2)}$

# Output interaction prediction using feed forward network (FFN)
$y_{\text{out}} \leftarrow \texttt{SoftMax}\left(\text{FFN}\left(\texttt{concat}\left[s^{(1)}, s^{(2)}\right]\right)\right)$

return Prediction probability: $y_{\text{out}}$, Residue importance: $(p^{(1)}, p^{(2)})$
```

Results

As part of our assessment, we compare the performance of Struct2Graph for PPI predictions against a number of recent machine learning models. These methods include: (a) DeepFE model [30], where we train the natural language processing network on the same database used in the original publication and feed the embeddings into a fully connected feedforward neural network. (b) DeepPPI [29], where we extract 1164 sequence features related to the amino acid composition, distribution, and order. A separate neural network is used for each protein in the protein–protein pair and their outputs are concatenated into a final network for classification. Furthermore, as was done in the original publication [29], we implement these features into a number of traditional machine learning models [70], such as (c) Gaussian naive Bayes (GaussianNB) classifier, (d) Quadratic discriminant analysis (QDA), (e) k-nearest neighbor (k-NN) classifier, (f) Decision tree (DT) classifier, (g) Random forest (RF) classifier, (h) Adaboost classifier, and (i) Support vector classifier (SVC) [71]. All models are implemented in Python 3.6.5 on an Intel i7-7700HQ CPU with 2.8 GHz x64-based processor. For common machine learning classifiers, such as, GaussianNB, QDA, SVC, RF, DT, k-NN and Adaboost, we use the readily available implementation in the scikit-learn [70] module. Deep learning classifiers, in particular, DeepPPI [72] and DeepFE-PPI [73] are implemented in Keras [74], while Struct2Graph is implemented in PyTorch [75].

For Struct2Graph, the hyperparameters of the models are tuned in order to achieve the reported accuracies. The tuning is obtained by performing grid search over the set of possible hyperparameter settings. The hyperparameters of our Struct2Graph implementation are as follows: `optimizer`: Adam optimizer [76] with learning rate $\lambda = 10^{-3}$ and rate-decay of 0.5 per 10 epochs; `total epochs`: 50; `number of GCN layers`: $l = 2$; `GCN embedding dimension`: $d = 20$; `loss function`: binary cross-entropy. For other competing methods, we use the tuned hyperparameters that are adopted from the original publications.

Performance on balanced databases

Table 1 summarizes the comparisons of Struct2Graph and various machine learning models for PPI prediction for a fivefold stratified cross validation study. In the cross validation, the 10,004 pairs (4698 positive and 5036 negatives) are randomly partitioned into five subsamples of equal size. Of these five subsamples, a single subsample is retained as the validation data for testing various machine learning models, and the remaining four subsamples are used as training data. In order to reduce the training time with our Struct2Graph model, 800 pairs are randomly sampled with replacement among the 8003 pairs (80%) in each epoch, and the performance on the randomly chosen 800 pairs is used to update the parameters of the neural network. This modification not only reduces the training time considerably, but also injects noise into the training data to avoid any potential overfitting.

The performance is reported for various measures, such as, accuracy, precision, recall, specificity or the true negative rate, Matthews correlation coefficient (MCC), F_1 -score, area under the receiver operating characteristic curve (ROC-AUC), and negative predictive value (NPV) (see Tables 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10). For a balanced

Table 1 Fivefold cross-validation performance analysis of several machine learning methods on balanced dataset (1:1)

Method	**Performance (%) on balanced training set—1:1**			
	Accuracy	**Precision**	**Recall**	**Specificity**
GaussianNB	72.14 ± 2.91	98.41 ± 0.51	45.05 ± 6.10	99.24 ± 0.30
QDA	78.66 ± 3.44	70.43 ± 3.41	**99.42 ± 0.40**	57.90 ± 7.12
k-NN	94.19 ± 0.56	99.49 ± 0.08	88.83 ± 1.10	99.54 ± 0.07
Decision trees	96.20 ± 0.43	97.59 ± 0.28	94.75 ± 0.99	97.66 ± 0.29
Random forest	98.86 ± 0.29	99.45 ± 0.19	98.27 ± 0.49	99.45 ± 0.19
Adaboost	97.85 ± 0.26	98.76 ± 0.18	96.92 ± 0.51	98.78 ± 0.18
SVC	98.49 ± 0.33	99.44 ± 0.18	97.53 ± 0.61	99.45 ± 0.18
DeepPPI	97.22 ± 0.44	98.26 ± 0.82	96.14 ± 0.88	98.29 ± 0.83
DeepFE-PPI	98.64 ± 0.32	99.16 ± 0.28	98.12 ± 0.51	99.17 ± 0.28
Struct2Graph	**98.89 ± 0.24**	**99.50 ± 0.36**	98.37 ± 0.34	**99.45 ± 0.42**
Method	**MCC**	**F1-score**	**ROC-AUC**	**NPV**
GaussianNB	52.69 ± 4.38	61.53 ± 6.00	95.24 ± 0.33	64.46 ± 2.37
QDA	63.06 ± 5.23	82.40 ± 2.27	78.66 ± 3.43	**99.05** ± 0.58
k-NN	88.89 ± 1.02	93.86 ± 0.63	98.79 ± 0.21	89.92 ± 0.89
Decision trees	92.45 ± 0.84	96.15 ± 0.46	96.36 ± 0.30	95.23 ± 0.61
Random forest	97.74 ± 0.58	98.86 ± 0.30	**99.63 ± 0.08**	98.30 ± 0.46
Adaboost	95.72 ± 0.52	97.83 ± 0.27	99.20 ± 0.08	96.94 ± 0.50
SVC	97.01 ± 0.66	98.48 ± 0.34	**99.63 ± 0.09**	97.58 ± 0.59
DeepPPI	94.47 ± 0.87	97.19 ± 0.44	99.28 ± 0.11	96.23 ± 0.81
DeepFE-PPI	97.29 ± 0.64	98.64 ± 0.32	99.52 ± 0.09	98.14 ± 0.50
Struct2Graph	**97.79 ± 0.49**	**98.94 ± 0.20**	99.55 ± 0.16	98.24 ± 0.42

Bold face numbers indicate the best performance

Note that the proposed Struct2Graph method outperforms all other methods on the majority of metrics

training set (Table 1), Struct2Graph outperforms any other existing machine learning models in the literature for all the measures (except for the recall, NPV, and ROC-AUC scores) with an average accuracy and precision of 98.89% and 99.50%, respectively. This is despite the fact that we significantly downsample the number of pairs in each epoch during the training process of the proposed Struct2Graph model.

Note from Table 1 that while QDA outperforms Struct2Graph in terms of recall and NPV scores, it does very poorly in terms of other measures indicating that the QDA classifier overestimates positive interactions resulting in high false positive counts. Another observation is that the performance of Struct2Graph is only slightly better than that of another deep learning PPI model, DeepFE-PPI for this balanced training set. However, as discussed below, DeepFE-PPI does not perform as well for unbalanced training set, where positive interactions are underrepresented among all interactions, a case that often arises in practice.

The primary purpose of *k*-fold cross validation study is to measure the generalization capabilities of a model. Bootstrap resampling, on the other hand, is primarily used to establish empirical distribution functions for a widespread range of statistics. It works by performing sampling with replacement from the original dataset, and at the same time assuming that the data points that have not been chosen, are the test dataset. We repeat this procedure several times and compute the average score as

Table 2 Bootstrap resampling performance analysis of several machine learning methods on balanced dataset (1:1)

Method	**Performance (%) on Balanced training set—1:1**			
	Accuracy	**Precision**	**Recall**	**Specificity**
GaussianNB	71.90 ± 3.19	99.01 ± 0.38	46.39 ± 5.95	99.47 ± 0.23
QDA	54.46 ± 0.93	53.28 ± 0.77	**99.96 ± 0.06**	5.30 ± 1.74
k-NN	92.77 ± 0.22	99.38 ± 0.13	86.62 ± 0.50	99.41 ± 0.11
Decision trees	95.15 ± 0.51	97.13 ± 0.38	93.41 ± 0.88	97.03 ± 0.31
Random forest	98.70 ± 0.11	99.39 ± 0.18	98.10 ± 0.14	99.35 ± 0.18
Adaboost	96.61 ± 0.25	97.83 ± 0.40	95.60 ± 0.21	97.71 ± 0.44
SVC	98.05 ± 0.19	99.35 ± 0.15	96.87 ± 0.28	99.32 ± 0.15
DeepPPI	97.16 ± 0.40	98.04 ± 0.96	96.27 ± 0.84	98.06 ± 0.98
DeepFE-PPI	98.41 ± 0.12	98.81 ± 0.50	98.11 ± 0.45	98.73 ± 0.55
Struct2Graph	**98.96 ± 0.19**	**99.40 ± 0.09**	98.57 ± **0.35**	**99.47 ± 0.09**
Method	**MCC**	**F1-score**	**ROC-AUC**	**NPV**
GaussianNB	53.40 ± 4.44	62.93 ± 5.81	96.04 ± 0.12	63.30 ± 2.72
QDA	16.43 ± 2.53	69.50 ± 0.66	52.63 ± 0.84	**99.37 ± 0.83**
k-NN	86.36 ± 0.40	92.56 ± 0.30	98.32 ± 0.16	87.31 ± 0.38
Decision trees	95.23 ± 0.55	95.27 ± 0.54	94.21 ± 0.71	93.18 ± 0.81
Random forest	97.41 ± 0.22	98.74 ± 0.11	99.58 ± 0.07	97.97 ± 0.17
Adaboost	93.25 ± 0.50	96.70 ± 0.23	99.00 ± 0.09	95.36 ± 0.29
SVC	96.13 ± 0.37	98.10 ± 0.20	99.53 ± 0.09	96.71 ± 0.28
DeepPPI	94.36 ± 0.81	97.14 ± 0.40	99.05 ± 0.14	96.34 ± 0.78
DeepFE-PPI	96.82 ± 0.25	98.45 ± 0.12	99.52 ± 0.05	97.99 ± 0.20
Struct2Graph	**97.91 ± 0.38**	**98.98 ± 0.19**	**99.62 ± 0.17**	98.50 ± 0.33

Bold face numbers indicate the best performance

Note that the proposed Struct2Graph method outperforms all other methods on the majority of metrics

Table 3 Fivefold cross-validation performance analysis of deep-learning based machine learning methods on unbalanced dataset (1:2)

Method	**Performance (%) on unbalanced training set—1:2**			
	Accuracy	**Precision**	**Recall**	**Specificity**
DeepPPI	97.40 ± 0.44	98.64 ± 0.61	93.52 ± 1.64	99.35 ± 0.30
DeepFE-PPI	98.91 ± 0.09	99.00 ± 0.32	97.71 ± 0.33	99.51 ± 0.16
Struct2Graph	**99.03 ± 0.24**	**99.13 ± 0.25**	**98.11 ± 0.58**	**99.53 ± 0.13**
Method	**MCC**	**F1-score**	**ROC-AUC**	**NPV**
DeepPPI	94.16 ± 0.97	96.00 ± 0.72	99.19 ± 0.21	96.85 ± 0.76
DeepFE-PPI	97.54 ± 0.20	98.35 ± 0.13	**99.56 ± 0.08**	**99.86 ± 0.16**
Struct2Graph	**97.87 ± 0.51**	**98.62 ± 0.32**	99.47 ± 0.20	98.97 ± 0.34

Bold face numbers indicate the best performance

estimation of the performances of various classifiers. Table 2 summarizes the comparisons of Struct2Graph and various machine learning models on a balanced dataset for PPI prediction for a bootstrap resampling method repeated over five times. As before, we downsample the number of pairs in each epoch during the training process of the Struct2Graph in order to speed up computation and avoid any potential

Table 4 Fivefold cross-validation performance analysis of deep-learning based machine learning methods on unbalanced dataset (1:3)

Method	Performance (%) on unbalanced training set—1:3			
	Accuracy	Precision	Recall	Specificity
DeepPPI	98.19 ± 0.58	98.73 ± 0.40	93.98 ± 2.43	99.59 ± 0.13
DeepFE-PPI	98.96 ± 0.27	98.30 ± 0.46	97.52 ± 0.88	99.44 ± 0.15
Struct2Graph	**99.30 ± 0.22**	**99.17 ± 0.44**	**98.19 ± 1.09**	**99.71 ± 0.13**
Method	MCC	F1-score	ROC-AUC	NPV
DeepPPI	95.15 ± 1.55	96.28 ± 1.24	99.27 ± 0.14	98.03 ± 0.78
DeepFE-PPI	97.21 ± 0.72	97.90 ± 0.55	**99.51 ± 0.11**	99.18 ± 0.29
Struct2Graph	**98.20 ± 0.56**	**98.67 ± 0.41**	99.49 ± 0.21	**99.33 ± 0.38**

Bold face numbers indicate the best performance

Table 5 Fivefold cross-validation performance analysis of deep-learning based machine learning methods on unbalanced dataset (1:5)

Method	Performance (%) on unbalanced training set—1:5			
	Accuracy	Precision	Recall	Specificity
DeepPPI	97.78 ± 0.45	98.33 ± 0.39	88.20 ± 2.74	**99.70 ± 0.07**
DeepFE-PPI	98.97 ± 0.27	98.19 ± 0.49	95.60 ± 1.52	99.65 ± 0.10
Struct2Graph	**99.13 ± 0.18**	**98.49 ± 0.85**	**96.63 ± 0.93**	99.68 ± 0.19
Method	MCC	F1-score	ROC-AUC	NPV
DeepPPI	91.87 ± 1.68	92.97 ± 1.53	98.69 ± 0.50	97.69 ± 0.52
DeepFE-PPI	96.28 ± 1.00	96.87 ± 0.85	**99.56 ± 0.25**	99.12 ± 0.30
Struct2Graph	**97.03 ± 0.56**	**97.55 ± 0.45**	99.17 ± 0.25	**99.26 ± 0.23**

Bold face numbers indicate the best performance

Table 6 Fivefold cross-validation performance analysis of deep-learning based machine learning methods on unbalanced dataset (1:10)

Method	Performance (%) on Unbalanced training set—1:10			
	Accuracy	Precision	Recall	Specificity
DeepPPI	98.24 ± 0.49	95.83 ± 2.60	84.33 ± 4.23	99.63 ± 0.23
DeepFE-PPI	99.17 ± 0.33	96.56 ± 1.09	94.19 ± 2.87	99.67 ± 0.10
Struct2Graph	**99.42 ± 0.14**	**97.54 ± 1.28**	**96.43 ± 2.49**	**99.73 ± 0.16**
Method	MCC	F1-score	ROC-AUC	NPV
DeepPPI	88.95 ± 3.14	89.66 ± 2.94	97.18 ± 1.24	98.45 ± 0.41
DeepFE-PPI	94.91 ± 2.07	95.35 ± 1.90	**99.48 ± 0.32**	99.42 ± 0.29
Struct2Graph	**96.65 ± 1.12**	**96.96 ± 1.07**	99.45 ± 0.70	**99.63 ± 0.22**

Bold face numbers indicate the best performance

overfitting. The performance statistics for the Struct2Graph method with bootstrap resampling are very similar to the ones obtained with a fivefold cross-validation study. Struct2Graph is shown to outperform other existing machine learning models for all the measures (except for the recall and NPV scores) with an average accuracy and precision of 98.96% and 99.40%, respectively. Interestingly, the performances of the

Table 7 Bootstrap resampling performance analysis of deep-learning based machine learning methods on unbalanced dataset (1:2)

Method	**Performance (%) on Unbalanced training set—1:2**			
	Accuracy	**Precision**	**Recall**	**Specificity**
Decision trees	94.86 ± 0.58	95.67 ± 1.29	89.48 ± 0.79	97.79 ± 0.67
Random forest	98.74 ± 0.15	99.10 ± 0.17	97.33 ± 0.44	99.45 ± 0.09
DeepPPI	97.91 ± 0.38	98.37 ± 0.69	95.32 ± 1.40	99.21 ± 0.34
DeepFE-PPI	98.53 ± 0.20	98.41 ± 0.62	97.37 ± 0.31	99.16 ± 0.31
Struct2Graph	**98.91 ± 0.24**	**99.17 ± 0.15**	**97.89 ± 0.17**	**99.52 ± 0.27**
Method	**MCC**	**F1-score**	**ROC-AUC**	**NPV**
Decision trees	88.69 ± 1.28	92.47 ± 0.82	93.66 ± 0.58	94.47 ± 0.50
Random forest	97.25 ± 0.33	98.20 ± 0.22	99.71 ± 0.08	98.56 ± 0.24
DeepPPI	95.29 ± 0.85	96.81 ± 0.60	99.29 ± 0.25	97.70 ± 0.67
DeepFE-PPI	96.76 ± 0.45	97.88 ± 0.31	99.41 ± 0.17	98.59 ± 0.09
Struct2Graph	**97.59 ± 0.51**	**98.43 ± 0.30**	**99.73 ± 0.18**	**98.87 ± 0.16**

Bold face numbers indicate the best performance

Table 8 Bootstrap resampling performance analysis of deep-learning based machine learning methods on unbalanced dataset (1:3)

Method	**Performance (%) on Unbalanced training set—1:3**			
	Accuracy	**Precision**	**Recall**	**Specificity**
Decision trees	95.72 ± 0.46	95.55 ± 0.74	87.98 ± 1.63	98.52 ± 0.26
Random forest	98.80 ± 0.14	98.29 ± 0.33	97.15 ± 0.34	99.39 ± 0.11
DeepPPI	97.78 ± 0.36	98.43 ± 0.42	92.60 ± 1.38	99.51 ± 0.13
DeepFE-PPI	98.86 ± 0.11	98.42 ± 0.54	97.30 ± 0.31	99.43 ± 0.20
Struct2Graph	**99.01 ± 0.16**	**98.83 ± 0.37**	**97.42 ± 0.51**	**99.59 ± 0.13**
Method	**MCC**	**F1-score**	**ROC-AUC**	**NPV**
Decision trees	88.88 ± 1.14	91.60 ± 0.88	93.26 ± 0.81	95.78 ± 0.61
Random forest	96.90 ± 0.39	97.72 ± 0.29	99.72 ± 0.05	98.98 ± 0.10
DeepPPI	94.04 ± 0.98	95.42 ± 0.77	99.06 ± 0.47	97.58 ± 0.44
DeepFE-PPI	97.08 ± 0.28	97.85 ± 0.20	99.34 ± 0.11	99.03 ± 0.05
Struct2Graph	**97.46 ± 0.42**	**98.12 ± 0.32**	**99.75 ± 0.20**	**99.08 ± 0.18**

Bold face numbers indicate the best performance

DeepPPI and the DeepFE-PPI methods are marginally worse than that of the Random Forest classifier on the balanced set. However, as the class imbalance increases, DeepFE-PPI is shown to outperform the Random Forest classifier. We have, thus, also included the Random Forest classifier for comparative analysis on the unbalanced datasets.

Performance on unbalanced database

In most practical scenarios, the number of negative pairs is expected to be larger than positive pairs, since only a small fraction of protein pairs interact within all possible pairs. We thus evaluate the performance of the deep learning models, Deep-PPI and DeepFE-PPI against the proposed Struct2Graph model on various unbalanced training sets, where the number of negative pairs outnumber the positive pairs. These results are

Table 9 Bootstrap resampling performance analysis of deep-learning based machine learning methods on unbalanced dataset (1:5)

Method	**Performance (%) on Unbalanced training set—1:5**			
	Accuracy	**Precision**	**Recall**	**Specificity**
Decision trees	95.56 ± 0.45	94.14 ± 0.79	80.91 ± 2.12	98.87 ± 0.16
Random forest	98.38 ± 0.19	98.25 ± 0.65	92.82 ± 1.07	99.63 ± 0.13
DeepPPI	97.96 ± 0.46	97.13 ± 2.71	90.58 ± 3.21	99.44 ± 0.57
DeepFE-PPI	98.90 ± 0.31	98.20 ± 0.29	95.64 ± 1.75	99.61 ± 0.07
Struct2Graph	**99.16 ± 0.17**	**98.29 ± 0.64**	**97.08 ± 1.01**	**99.69 ± 0.13**
Method	**MCC**	**F1-score**	**ROC-AUC**	**NPV**
Decision trees	84.72 ± 1.33	87.01 ± 1.18	89.89 ± 1.04	95.82 ± 0.55
Random forest	94.53 ± 0.70	95.45 ± 0.61	99.66 ± 0.15	99.03 ± 0.11
DeepPPI	92.58 ± 1.66	93.66 ± 1.46	98.96 ± 0.13	98.15 ± 0.61
DeepFE-PPI	96.24 ± 1.11	96.89 ± 0.94	99.49 ± 0.11	99.05 ± 0.16
Struct2Graph	**97.03 ± 0.51**	**97.53 ± 0.41**	**99.71 ± 0.26**	**99.40 ± 0.23**

Bold face numbers indicate the best performance

Table 10 Bootstrap resampling performance analysis of deep-learning based machine learning methods on unbalanced dataset (1:10)

Method	**Performance (%) on Unbalanced training set - 1:10**			
	Accuracy	**Precision**	**Recall**	**Specificity**
Decision trees	96.63 ± 0.46	91.66 ± 1.47	73.14 ± 2.93	99.26 ± 0.14
Random forest	97.85 ± 0.25	95.87 ± 0.68	82.12 ± 2.28	99.61 ± 0.06
DeepPPI	98.09 ± 0.77	95.30 ± 3.86	83.29 ± 7.49	99.58 ± 0.37
DeepFE-PPI	98.50 ± 0.46	96.56 ± 0.38	87.86 ± 5.00	99.66 ± 0.05
Struct2Graph	**99.26 ± 0.15**	**97.04 ± 0.70**	**95.59 ± 0.73**	**99.67 ± 0.10**
Method	**MCC**	**F1-score**	**ROC-AUC**	**NPV**
Decision trees	80.13 ± 2.06	81.33 ± 1.99	86.20 ± 1.48	97.07 ± 0.45
Random forest	87.60 ± 1.12	88.44 ± 1.11	99.49 ± 0.22	98.03 ± 0.32
DeepPPI	88.01 ± 4.96	88.69 ± 4.82	96.65 ± 1.60	98.35 ± 0.73
DeepFE-PPI	91.27 ± 2.80	91.92 ± 2.72	99.50 ± 0.20	98.69 ± 0.24
Struct2Graph	**95.90 ± 0.60**	**96.31 ± 0.52**	**99.54 ± 0.22**	**99.50 ± 0.12**

Bold face numbers indicate the best performance

summarized in Tables 3, 4, 5 and 6 for several databases with varying ratios of positive to negative pairs: (a) 1:2 (2518 positive and 5036 negative), (b) 1:3 (1679 positive and 5036 negative), (c) 1:5 (1007 positive and 5036 negative), and (d) 1:10 (504 positive and 5036 negative). Note that the positive pairs for unbalanced databases are selected randomly from the set of curated positive pairs. Struct2Graph again outperforms its deep-learning counterparts consistently for this unbalanced case. Struct2Graph improvement increases when the ratio between positive and negative pairs becomes increasingly skewed. For instance, when the ratio of positive and negative pairs is 1:10, the precision and recall statistics for the Struct2Graph model are 97.54% and 96.43%, respectively, which are higher by 0.98% and 2.14%, respectively than the performance of the next best deep-learning model, DeepFE-PPI.

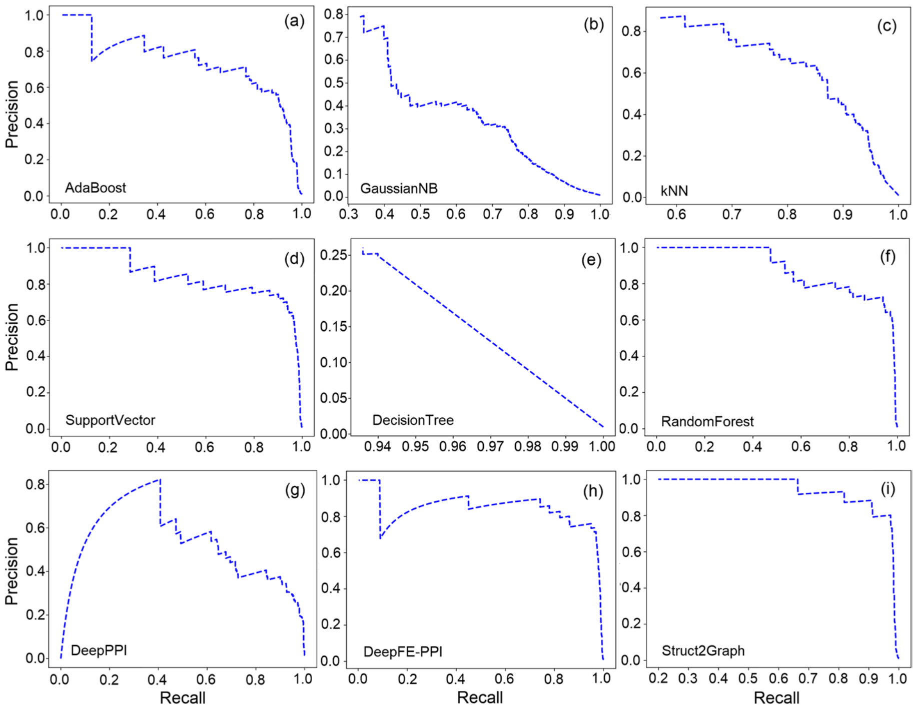

Fig. 3 Prevalence corrected precision-recall curves for the balanced database. **a** AdaBoost classifier, **b** GaussianNB classifier, **c** *k*NN classifier, **d** SVC, **e** Decision tree classifier, **f** Random forest classifier, **g** DeepPPI classifier, **h** DeepFE-PPI classifier, **i** Struct2Graph (ours) classifier

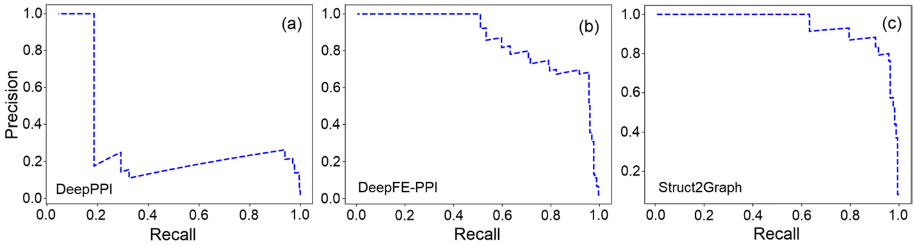

Fig. 4 Prevalence corrected precision-recall curves for the unbalanced database. **a** DeepPPI classifier, **b** DeepFE-PPI classifier, **c** Struct2Graph (ours) classifier

Bootstrap resampling yields a very similar conclusion, where the Struct2Graph is again shown to outperform its deep-learning counterparts, as well as the Random Forest classifier, consistently for several unbalanced cases (see Tables 7, 8, 9 and 10). When the ratio of positive and negative pairs is 1:10, the accuracy, precision and recall statistics for the Struct2Graph model are 99.26%, 97.04% and 95.59%, respectively, which are higher by 0.76%, 0.58% and 7.73%, respectively than the performance of the next best deep-learning model, DeepFE-PPI.

While a ratio of 1:10 reflects a significant class imbalance between positive and negative examples, the class imbalance in a protein interactome can potentially be of the order of 1:100 or even larger. In the absence of a PPI database (consisting of 3D-structural information) with such huge class imbalance, prevalence-corrected

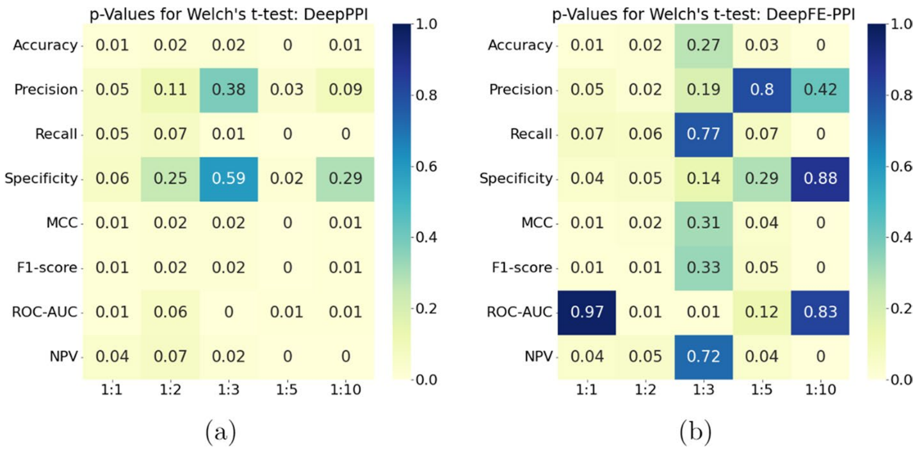

Fig. 5 *p*-value statistical significance when Struct2Graph is compared with **a** DeepPPI, **b** DeepFE-PPI, using the Welch's *t*-test. The columns depict scenarios in which the model was trained beginning from the balanced (1:1), to unbalanced (1:2, 1:3, 1:5, 1:10) datasets

Precision-Recall Curves (PRCs) have been adopted [77] that reduce the false positive rate at the expense of the true positive rate. Figure 3 depicts the prevalence-corrected PRCs on the balanced (1:1) for several PPI classifiers. The computation of precision is suitably modified with $r = 100$ [77] for an expected ratio of 1:100 for positive to negative samples in the real-world data. PRCs best summarize the trade-off between the true positive rate and the positive predictive value (PPV) for a classifier using different probability thresholds. The AUC (area under curve) in Fig. 3i nearly approaches unity, thus guaranteeing excellent discrimination capability of the proposed Struct2Graph architecture. Figure 4 depicts the prevalence-corrected PRCs for the deep-learning classifiers on the unbalanced (1:10) dataset. As before, the AUC for the Struct2Graph architecture almost approaches unity.

Statistical test for comparing PPI prediction algorithms

Tables 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10 depict that the Struct2Graph outperforms other competing classifiers on (almost) all metrics. On the other hand, other deep-learning based classifiers, such as the DeepPPI and the DeepFE-PPI, do not seem to perform as well as the Struct2Graph classifier, they still manage to get reasonably close to Struct2Graph on several performance measures. We, thus, compare these two classifiers with the Struct2Graph classifier using the Welch's *t*-test to elucidate statistically significant evidence in favor of the Struct2Graph classifier. In particular, we compare the means of the proposed Struct2Graph model with the DeepPPI and the DeepFE-PPI models, respectively, across all folds of the cross-validation set for each metric using the one-sided Welch's *t*-test. Figure 5 depicts the *p*-value statistical significance for rejecting the null hypothesis that means are equal. Here, the rows represent several performance measures, while the columns depict scenarios in which the model was trained beginning from the balanced (1:1), to unbalanced (1:2, 1:3, 1:5, 1:10) datasets. Recall that a *p*-value less than 0.05 is typically considered to be statistically significant. It can be seen that there is statistically significant evidence in favor of superior performance of the Struct2Graph classifier.

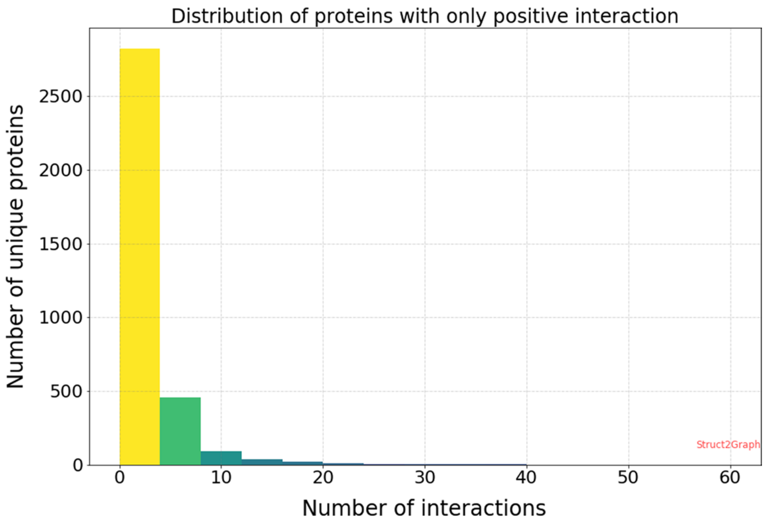

Fig. 6 Histogram of proteins with only positive interactions. Of the 3677 unique PDBs, 3453 PDBs are involved in only positive interactions, i.e., among all the protein–protein pair instances in our database, these 3453 proteins do not feature in any non-complex forming instance. Moreover, of the 3453 PDBs with only positive interactions, nearly 82% unique PDBs are involved in fewer than 4 PPI examples. Consequently, for a classifier to memorize data and not "learn" to predict interactions would be extremely difficult without each PDB appearing in a relatively large number of PPI instances

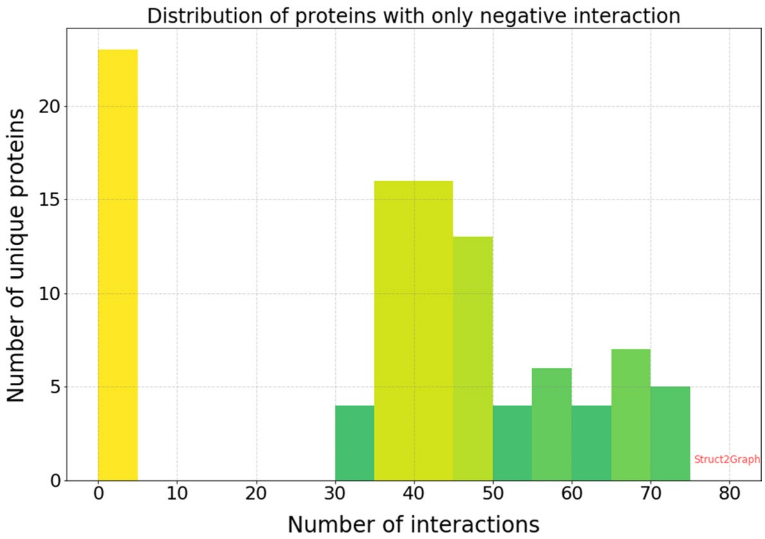

Fig. 7 Histogram of proteins with only negative interactions. Of the 3677 unique PDBs, only 104 PDBs are involved in just the negative interactions, i.e., among all the protein–protein pair instances in our database, these 104 proteins do not feature in any complex forming instance. Moreover, of the 104 PDBs, 23 PDBs appear in less than 5 PPI examples. The total number of proteins that are involved in more than 5 PPI examples is a very small number (81), i.e., only 2.2% of the entire PDB database considered in our work

Heterogeneity of the PPI database

The success of any machine learning algorithm is based on the quality of training data it is presented with. Of the 4698 positive and 5036 negative examples spanned across 3677 unique proteins, we first want to make sure that the learning algorithms are not biased towards memorizing the training data [78], since some of the proteins in the database

Table 11 Memorization test for the PPI database

Database	Accuracy on train-set (%)	Accuracy on test-set (%)
Balanced (1:1)	91.17	91.07
Unbalanced (1:2)	66.98	50.43
Unbalanced (1:3)	75.25	50.23
Unbalanced (1:5)	83.53	50.43
Unbalanced (1:10)	91.04	50.33

are involved in positive only or negative only interactions. Figure 6 shows the distribution of number of interactions per protein that are involved in positive only interactions. It can be seen that nearly 82% of the unique proteins are involved in four or fewer positive only interactions. Consequently, for a classifier to memorize the training data and not learn to predict positive interactions would be extremely difficult without each protein appearing in a relatively large number of PPI instances.

Similarly, of the 3677 unique proteins, 104 unique proteins are involved in negative only interactions (do not form complexes). Figure 7 shows the distribution of number of interactions per protein that are involved in negative only interactions. As seen in the histogram, the total number of proteins with negative only interactions appearing in more than five PPI examples is very small (81), and comprise of only 2.2% of the entire PDB database considered in our work. Thus, the distribution of data makes it implausible for learning algorithms to perform well just by memorizing the training data. The dataset also consists of 120 unique proteins that are involved in both positive and negative interactions. These 120 unique proteins appear in 6335 PPI instances in our database. Hence, it would be nearly impossible for any classifier to simply memorize the training data, and still be able to predict the interactions almost accurately on the test or validation set.

We further validate this by building a random forest classifier that is supplied with an input vector of length 3677. The specific choice of length of the input vector is directly related to the number of unique proteins in the database. We first create a dictionary of all the unique proteins and note down the order in which these unique proteins appear in the dictionary. Then, each unique protein is represented by a 3677-long unit vector with all but one coordinate being zero. The coordinate corresponding to the order of the protein in the dictionary is marked as 1. For predicting the interaction of two proteins, say proteins A and B, the unit vectors are summed and supplied to the random forest classifier as an input. Recall that the sum operation is permutation invariant, and thus interaction prediction for the pair (protein A, protein B) is identical to that for the pair (protein B, protein A). Table 11 summarizes the performance of the random forest classifiers on balanced and unbalanced datasets, trained using only the labels of protein pairs and disregarding any structural information. In the balanced scenario, the classifier can be trained with a reasonable accuracy of $\sim 91\%$ on both training and test sets. This is still significantly smaller than accuracies obtained using Struct2Graph and other deep-learning based classifiers on the balanced set. However, as the training database is made more realistic (i.e., biased towards significantly abundant negative examples), the performance on the training set drops, while the performance on the test set is completely random ($\sim$ 50%), i.e., the random forest classifier acts like a random predictor. In the extreme scenario, (1:10 ration between positive and negative examples), the training accuracy appears improved, largely because every time the classifier predicts a negative interaction, it is likely to be correct since the training set has an abundance of negative examples. However, on the test set comprising of approximately equal number of positive or negative examples, the prediction accuracy is still around 50% indicating zero learning.

Discussion

The success of Struct2Graph is attributed to the thorough analysis of structural 3D information embedded in the form of a graph, which predicts interactions better than sequence-based approaches [13]. In addition, Struct2Graph can potentially identify residues that likely contribute to the formation of the protein–protein complex. This is achieved by considering the probability tuples $\{(p_i, p_j)\}$ of different amino acids during the knowledge selection process described in Eq. (3). These probabilities capture the relative importance of amino acids and thus reflect different amino acids' contributions towards interaction prediction. The amino acids with large relative probabilities (top 20%) are identified as important for the formation of the protein–protein complex. This importance can stem from either direct participation in the interaction process (i.e., binding site) or indirectly through contribution to appropriate protein folding that allows formation of the correct binding site geometry.

A demonstration of the potential of Struct2Graph to identify specific interaction sites was performed on two example cases (neither part of the training set) with well-described interacting residues from protein pairs in the literature. Specifically, we studied two different interaction types: (1) A protein with multiple ligands competing for the same binding area [79]; and (2) A dynamic protein–protein adhesion interaction [80]. The reported interacting residues in these complexes are compared with the Struct2Graph's highest probability residues (top 20%) using standard 2x2 confusion matrices. In aggregate (i.e., two case examples with a total of three interacting pairs), Struct2Graph identifies interacting residues with 30% sensitivity, 89% specificity, and 87% accuracy. It should be noted that these protein pair examples are not in the training set, and Struct2Graph identifies these residues through its knowledge selection process in a completely unsupervised manner. Besides, as noted above, the identified residues could be critical for ensuring correct protein folding conformation and therefore *indirectly* important for predicting binding, but not captured by traditional analysis that focuses only on the specific interacting residues identified in the literature. Detailed results for each example are described:

(1) *HMGB1 and PSMα_1 compete for binding TLR4*: Phenol soluble modulins (PSMs), short, amphipathic, helical peptides [81], play a crucial role in *Staphylococcus aureus* virulence, one of the most common causes of human bacterial infections worldwide [82]. *S. aureus* has seven PSMs (PSM$\alpha_1 - \alpha_4$, PSM$\beta_1 - \beta_2$, and δ-toxin) which have multiple functions including, cytolysis, biofilm structuring, and inflam-

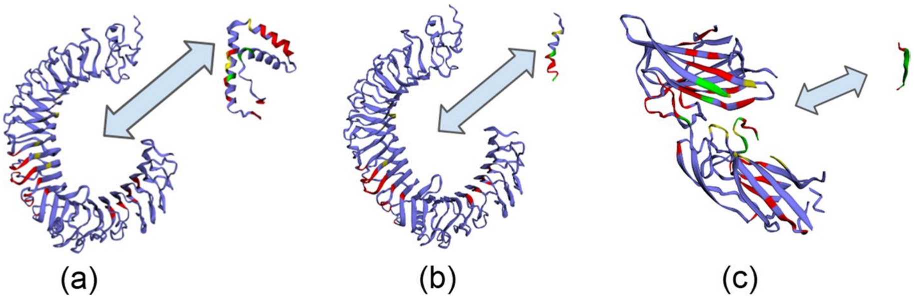

Fig. 8 Important residue prediction by Struct2Graph for three example scenarios. **a** TLR4 with HMGB1, **b** TLR4 with PSMα_1, **c** SdrG and Fibrinogen adhesion. The different colored residues encode different information: (i) Red: Top-20% residues identified important by Struct2Graph, (ii) Yellow: Actual binding site not identified to be important by Struct2Graph, (iii) Green: True binding site overlapping with a residue identified important by Struct2Graph, (iV) Purple: neither important, nor actual interaction site. Recall that both HMGB1 and PSMα_1 are known to compete for the same binding sites on TLR4, and this is reflected in the Struct2Graph predictive analysis as well

matory activation via cytokine release and chemotaxis. PSMs specifically trigger the release of high mobility group box-1 protein (HMGB1). Toll-like receptor-4 (TLR4) interacts with HMGB1 activating nuclear factor NF-κB and proinflammatory cytokines production [83]. However, *S. aureus* PSMs$\alpha_1 - \alpha_3$ significantly inhibit HMGB1-mediated phosphorylation of NF-κB by competing with HMGB1 via interactions with the same residues of TLR4 domain [79]. As such, the specific interacting residues for these pairs HMGB1:TLR4 (2LY4 : 3FXI) and PSMα_1:TLR4 (5KHB : 3FXI) have been well described [79].

Struct2Graph identifies interacting residues of the HMGB1:TLR4 pair with 90% accuracy in which the top 9 predicted residues for TLR4 fall within the reported active cavity (residues rank 336–477). In addition, among the top 20% predicted residues of HMGB1 were the specific interacting residues Tyr^{16} and Lys^{68}. For the PSMα_1:TLR4 pair, Struct2Graph identifies interacting residues with 92% accuracy. Again the top predicted residues fall within the previously identified TLR4 active cavity (336–477). For PSMα_1, interacting residues Gly^2 and Val^{10} were correctly identified. While the overall sensitivity for detecting an interacting residue is $\sim$ 20% for this example, Struct2Graph was able to predict that PSMα_1 interacts with TLR4 in the same area as the HMGB1 binding site. More specifically, the predicted binding sites for both on TLR4 have 94% concordance. Figure 8a and b shows the residues predicted to be essential and highlights how Struct2Graph predicts a similar site for both interactions.

(2) *SdrG-Fibrinogen Adhesion*: Microbial attachment to host tissues is a crucial step in most bacterial infections. Gram-positive pathogens such as Staphylococci, Streptococci, and Enterococci contain multiple cell wall-anchored proteins that act as an adhesin to mediate bacterial attachment to host tissues. These adhesin mediating interactions have been termed MSCRAMMs (microbial surface components recognizing adhesive matrix molecules) [84]. SdrG is an MSCRAMM of *Staphylococ-*

cus epidermidis that binds to the $B\beta$ chain of human fibrinogen (Fg) via dynamic "dock, lock, and latch" mechanism [80].

Struct2Graph was used to evaluate the interaction between SdrG (PDB:r19A) and a synthetic peptide with homologous sequence to its binding site in Fg (PDB:r17C). Interacting residues between SdrG and the synthetic Fg peptide homolog were predicted with 75% accuracy. Among the high probability residues identified in SdrG were 9 exact matches to those in the literature [80]. This included, Pro^{337}, Ser^{338}, Leu^{340}, Phe^{344}, Gln^{425}, Ser^{437}, Tyr^{577}, Asp^{578}, and Asn^{579}. Figure 8c shows the residues predicted to be essential for the interaction.

These results show that Struct2Graph provides insight into key residues involved in the protein–protein interaction without any training data on the specific nature of these interactions. A complete summary of the residues identified by Struct2Graph for the preceding examples is included in the supplementary material (see Additional file 1). Any high probability residues identified but not confirmed as directly interacting may have indirect effects through maintaining appropriate 3D conformation of the protein.

In addition to these specific binding examples, we consider the ability of our attention mechanism to predict useful residues across a broader dataset. Our attention mechanism does not necessarily predict interaction sites, but rather residues which are important to protein interactions regardless of their proximity to the interface. It has been observed that residues throughout the entire peptide chain can drive interactions [85]. Therefore, the attention mechanism will identify residues regardless of location that significantly alter interaction propensity. To demonstrate this, we analyze the single amino acid variation (SAV) dataset presented in [86] (please refer to the supporting information Additional file 2 for the human SAV dataset). The authors in [86] performed a large-scale structural analysis of human single amino acid variations (SAVs) and demonstrated that disease-causing mutations are preferentially located within the interface core, as opposed to the rim. Their work analyzed a total of 3282 disease-causing SAVs and 1699 benign polymorphisms occurring in 705 proteins. It is established that the disease-causing SAVs were 49% more likely to occur in the interface core rather than the rim and were 72% more likely to occur in the interface core than in the non-interacting protein surface, thus clearly demonstrating a different contribution of core and rim regions to human disease. On the other hand, 78.7% of polymorphisms were found to reside within surface-accessible residues (241 in interface residues and 1096 in surface non-interface residues), i.e, polymorphisms are less likely to be located in the interface core compared to the rim.

Since the work in [86] primarily dealt with human database, there is sufficient overlap between their dataset and the PPI database used in our manuscript. Of the overlapping 2724 disease-causing SAVs (spanning across 342 unique proteins) and 1364 polymorphisms (spanning across 528 unique proteins), our attention mechanism identifies 33.55% of all disease-causing SAVs as important (attention weights within top-20%), while 85.30% of all polymorphisms are identified as *un*important by the proposed attention mechanism, indicating significant overlap between the previously established SAV study and the important residues identified by the proposed attention mechanism.

Conclusion

Struct2Graph, a GCN-based mutual attention classifier, to accurately predict interactions between query proteins exclusively from 3D structural data is proposed. Since the prior study showed that the geometrical and graph theoretical descriptors may be sufficient for description of PPI [13], Struct2Graph does not directly use descriptors, such as sequence information, hydrophobicity, surface charge and solvent accessible surface area, and thus can be generalized to a broader class of nanoscale structures that can be represented in similar fashion. This study demonstrates that a relatively low-dimensional feature embedding learned from graph structures of individual proteins outperforms other modern machine learning classifiers based on global protein features. Our GCN-based classifier achieves state-of-the-art performance on both balanced and unbalanced datasets.

Moreover, the mutual attention mechanism provides insights into important residues that are likely to contribute towards interaction through direct or indirect participation. This is achieved through its knowledge selection process in a completely unsupervised manner. The identification of important residues is tested for two different interaction types: (a) Protein with multiple ligands competing for the same binding area, (b) Dynamic protein–protein adhesion interaction. Struct2Graph identifies interacting residues with 30% sensitivity, 89% specificity, and 87% accuracy. Finally, through the analysis of single amino acid variations, the attention mechanism shows preference for disease-causing residue variations over benign ones, demonstrating that it is not limited to interface residues. This connection between the unsupervised discovery of interaction sites and graph representation of proteins is possible thanks to the somewhat limited type of atoms and bond patterns that commonly occur in such molecules, which makes it possible to characterize properties on local atomistic arrangements. Overall, the proposed framework is general and, while subject to availability of corresponding training data, can be made to predict other kinds of complex sets of collective supramolecular interactions between proteins and nanoscale species of different chemical composition.

Abbreviations

PPI	Protein–protein interaction
GAT	Graph attention network
GCN	Graph convolutional network

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-022-04910-9.

Additional file 1: List of important residues as predicted by Struct2Graph.

Additional file 2: List of missense variants annotated in UniProtKB/Swiss-Prot human entries.

Acknowledgements

We thank the anonymous referees for their useful suggestions.

Author contributions

MB: Methodology, Software, Analysis, Writing—original draft. AM: Methodology, Software, Analysis. JS: Validation of PPI results, Comparison with existing methods. EST-E: Data curation, Validation, Writing—PPI database and interaction site prediction sections. SK: Data curation. PE: Methodology—representation and properties of molecules, Conceptualization, Writing—review and editing. JSV: Writing—review and editing, Supervision—PPI database. NAK: Methodology - graph representation of proteins and other nanostructures, concept of geometrical interactions in nano-bio structures, Writing—review and editing, Supervision—PPI database. AV: Conceptualization, Writing—review and editing, Supervision. AOH: Conceptualization, Writing—review and editing, Supervision. All authors read and approved the final manuscript.

Funding

AV, AOH, JSV, PE, MB, AM and SK acknowledge the support from the BlueSky Initiative from the University of Michigan College of Engineering. AOH acknowledges the support from ARO W911NF-19-1-0269 and ARO W911NF-14-1-0359. AV and JSV acknowledge the support from DARPA HR00111720067. NAK expresses gratitude to Vannewar Bush DoD Fellowship ONR N000141812876. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Availability of data and materials

The source code, along with the PPI database, is available at https://github.com/baranwa2/Struct2Graph.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Division of Data and Decision Sciences, Tata Consultancy Services Research, Mumbai, India. [2]Systems and Control Engineering Group, Indian Institute of Technology, Bombay, India. [3]Department of Computer Science, University of Albany, SUNY, Albany, USA. [4]Department of Chemical Engineering, University of Michigan, Ann Arbor, USA. [5]Department of Biomedical Engineering, University of Michigan, Ann Arbor, USA. [6]Department of Mechanical Engineering, University of Michigan, Ann Arbor, USA. [7]Department of Emergency Medicine, University of Michigan, Ann Arbor, USA. [8]Biointerfaces Institute, University of Michigan, Ann Arbor, USA. [9]Department of Materials Science and Engineering, University of Michigan, Ann Arbor, USA. [10]Biophysics Program, University of Michigan, Ann Arbor, USA. [11]Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, USA. [12]Department of Statistics, University of Michigan, Ann Arbor, USA. [13]Program in Applied Interdisciplinary Mathematics, University of Michigan, Ann Arbor, USA. [14]Program in Bioinformatics, University of Michigan, Ann Arbor, USA.

Received: 9 June 2022 Accepted: 26 August 2022
Published online: 10 September 2022

References

1. Berggård T, Linse S, James P. Methods for the detection and analysis of protein–protein interactions. Proteomics. 2007;7(16):2833–42. https://doi.org/10.1002/pmic.200700131.
2. Braun P, Gingras A-C. History of protein–protein interactions: from egg-white to complex networks. Proteomics. 2012;12(10):1478–98. https://doi.org/10.1002/pmic.201100563.
3. Phizicky EM, Fields S. Protein–protein interactions: methods for detection and analysis. Microbiol Rev. 1995;59(1):94–123. https://doi.org/10.1128/MMBR.59.1.94-123.1995.
4. Ito T, Chiba T, Ozawa R, Yoshida M, Hattori M, Sakaki Y. A comprehensive two-hybrid analysis to explore the yeast protein interactome. Proc Natl Acad Sci. 2001;98(8):4569–74. https://doi.org/10.1073/pnas.061034498.
5. Fry DC. Protein–protein interactions as targets for small molecule drug discovery. Biopolymers. 2006;84(6):535–52. https://doi.org/10.1002/bip.20608.
6. Coelho ED, Arrais JP, Luis-Oliveira J. From protein–protein interactions to rational drug design: Are computational methods up to the challenge? Curr Top Med Chem. 2013;13(5):602–18. https://doi.org/10.2174/1568026611313050005.
7. Mashaghi S, Jadidi T, Koenderink G, Mashaghi A. Lipid nanotechnology. Int J Mol Sci. 2013;14(2):4242–82.
8. Peppas NA, Huang Y. Nanoscale technology of mucoadhesive interactions. Adv Drug Deliv Rev. 2004;56(11):1675–87.
9. Lee S-M, Nguyen ST. Smart nanoscale drug delivery platforms from stimuli-responsive polymers and liposomes. Macromolecules. 2013;46(23):9169–80.
10. Meng H, Nel AE. Use of nano engineered approaches to overcome the stromal barrier in pancreatic cancer. Adv Drug Deliv Rev. 2018;130:50–7.
11. Kotov NA. Inorganic nanoparticles as protein mimics. Science. 2010;330(6001):188–9.
12. Bhandari S, Mondal D, Nataraj S, Balakrishna RG. Biomolecule-derived quantum dots for sustainable optoelectronics. Nanoscale Adv. 2019;1(3):913–36.
13. Cha M, Emre EST, Xiao X, Kim J-Y, Bogdan P, VanEpps JS, Violi A, Kotov NA. Unifying structural descriptors for biological and bioinspired nanoscale complexes. Nat Comput Sci. 2022;2(4):243–52.
14. Hu L, Wang X, Huang Y-A, Hu P, You Z-H. A survey on computational models for predicting protein–protein interactions. Brief Bioinform. 2021;22(5):036.

15. Uetz P, Giot L, Cagney G, Mansfield TA, Judson RS, Knight JR, Lockshon D, Narayan V, Srinivasan M, Pochart P, Qureshi-Emili A, Li Y, Godwin B, Conover D, Kalbfleisch T, Vijayadamodar G, Yang M, Johnston M, Fields S, Rothberg JM. A comprehensive analysis of protein–protein interactions in *Saccharomyces cerevisiae*. Nature. 2000;403(6770):623–7. https://doi.org/10.1038/35001009.
16. Gavin A-C, Bösche M, Krause R, Grandi P, Marzioch M, Bauer A, Schultz J, Rick JM, Michon A-M, Cruciat C-M, Remor M, Höfert C, Schelder M, Brajenovic M, Ruffner H, Merino A, Klein K, Hudak M, Dickson D, Rudi T, Gnau V, Bauch A, Bastuck S, Huhse B, Leutwein C, Heurtier M-A, Copley RR, Edelmann A, Querfurth E, Rybin V, Drewes G, Raida M, Bouwmeester T, Bork P, Seraphin B, Kuster B, Neubauer G, Superti-Furga G. Functional organization of the yeast proteome by systematic analysis of protein complexes. Nature. 2002;415(6868):141–7. https://doi.org/10.1038/415141a.
17. ...Ho Y, Gruhler A, Heilbut A, Bader GD, Moore L, Adams S-L, Millar A, Taylor P, Bennett K, Boutilier K, Yang L, Wolting C, Donaldson I, Schandorff S, Shewnarane J, Vo M, Taggart J, Goudreault M, Muskat B, Alfarano C, Dewar D, Lin Z, Michalickova K, Willems AR, Sassi H, Nielsen PA, Rasmussen KJ, Andersen JR, Johansen LE, Hansen LH, Jespersen H, Podtelejnikov A, Nielsen E, Crawford J, Poulsen V, Sorensen BD, Matthiesen J, Hendrickson RC, Gleeson F, Pawson T, Moran MF, Durocher D, Mann M, Hogue CWV, Figeys D, Tyers M. Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry. Nature. 2002;415:4.
18. Sprinzak E, Sattath S, Margalit H. How reliable are experimental protein–protein interaction data? J Mol Biol. 2003;327(5):919–23. https://doi.org/10.1016/S0022-2836(03)00239-0.
19. Skrabanek L, Saini HK, Bader GD, Enright AJ. Computational prediction of protein–protein interactions. Mol Biotechnol. 2008;38(1):1–17. https://doi.org/10.1007/s12033-007-0069-2.
20. Kaake RM, Wang X, Huang L. Profiling of protein interaction networks of protein complexes using affinity purification and quantitative mass spectrometry. Mol Cell Proteomics. 2010;9(8):1650–65.
21. Marcotte EM, Pellegrini M, Ng H-L, Rice DW, Yeates TO, Eisenberg D. Detecting protein function and protein–protein interactions from genome sequences. Science. 1999;285(5428):751–3. https://doi.org/10.1126/science.285.5428.751.
22. Sun J, Li Y, Zhao Z. Phylogenetic profiles for the prediction of protein–protein interactions: How to select reference organisms? Biochem Biophys Res Commun. 2007;353(4):985–91. https://doi.org/10.1016/j.bbrc.2006.12.146.
23. Pazos F, Valencia A. Similarity of phylogenetic trees as indicator of protein–protein interaction. Protein Eng Des Sel. 2001;14(9):609–14. https://doi.org/10.1093/protein/14.9.609.
24. Hashemifar S, Neyshabur B, Khan AA, Xu J. Predicting protein–protein interactions through sequence-based deep learning. Bioinformatics. 2018;34(17):802–10.
25. Zhang F, Song H, Zeng M, Li Y, Kurgan L, Li M. Deepfunc: a deep learning framework for accurate prediction of protein functions from protein sequences and interactions. Proteomics. 2019;19(12):1900019.
26. Shen J, Zhang J, Luo X, Zhu W, Yu K, Chen K, Li Y, Jiang H. Predicting protein–protein interactions based only on sequences information. Proc Natl Acad Sci. 2007;104(11):4337–41. https://doi.org/10.1073/pnas.0607879104.
27. Guo Y, Yu L, Wen Z, Li M. Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. Nucleic Acids Res. 2008;36(9):3025–30. https://doi.org/10.1093/nar/gkn159.
28. Mukherjee S, Zhang Y. Protein–protein complex structure predictions by multimeric threading and template recombination. Structure. 2011;19(7):955–66.
29. Du X, Sun S, Hu C, Yao Y, Yan Y, Zhang Y. DeepPPI: boosting prediction of protein–protein interactions with deep neural networks. J Chem Inf Model. 2017;57(6):1499–510. https://doi.org/10.1021/acs.jcim.7b00028.
30. Yao Y, Du X, Diao Y, Zhu H. An integration of deep learning with feature embedding for protein–protein interaction prediction. PeerJ. 2019;7:7126. https://doi.org/10.7717/peerj.7126.
31. Shi Q, Chen W, Huang S, Wang Y, Xue Z. Deep learning for mining protein data. Brief Bioinform. 2021;22(1):194–218.
32. Humphreys IR, Pei J, Baek M, Krishnakumar A, Anishchenko I, Ovchinnikov S, Zhang J, Ness TJ, Banjade S, Bagde SR, et al. Computed structures of core eukaryotic protein complexes. Science. 2021;374(6573):4805.
33. Liu L, Ma Y, Zhu X, Yang Y, Hao X, Wang L, Peng J. Integrating sequence and network information to enhance protein–protein interaction prediction using graph convolutional networks. In: 2019 IEEE international conference on bioinformatics and biomedicine (BIBM). IEEE; 2019. p. 1762–8.
34. Zhang D, Kabuka M. Multimodal deep representation learning for protein interaction identification and protein family classification. BMC Bioinform. 2019;20(16):1–14.
35. Yue X, Wang Z, Huang J, Parthasarathy S, Moosavinasab S, Huang Y, Lin SM, Zhang W, Zhang P, Sun H. Graph embedding on biomedical networks: methods, applications and evaluations. Bioinformatics. 2020;36(4):1241–51.
36. Yang F, Fan K, Song D, Lin H. Graph-based prediction of protein–protein interactions with attributed signed graph embedding. BMC Bioinform. 2020;21(1):1–16.
37. Huang K, Xiao C, Glass LM, Zitnik M, Sun J. Skipgnn: predicting molecular interactions with skip-graph networks. Sci Rep. 2020;10(1):1–16.
38. Rastogi C, Rube HT, Kribelbauer JF, Crocker J, Loker RE, Martini GD, Laptenko O, Freed-Pastor WA, Prives C, Stern DL, et al. Accurate and sensitive quantification of protein-DNA binding affinity. Proc Natl Acad Sci. 2018;115(16):3692–701.
39. Khandare J, Calderon M, Dagia NM, Haag R. Multifunctional dendritic polymers in nanomedicine: opportunities and challenges. Chem Soc Rev. 2012;41(7):2824–48.
40. Pelaz B, Alexiou C, Alvarez-Puebla RA, Alves F, Andrews AM, Ashraf S, Balogh LP, Ballerini L, Bestetti A, Brendel C, et al. Diverse applications of nanomedicine. ACS Nano. 2017;11(3):2313–81.
41. Xu L, Wang X, Wang W, Sun M, Choi WJ, Kim J-Y, Hao C, Li S, Qu A, Lu M, et al. Enantiomer-dependent immunological response to chiral nanoparticles. Nature. 2022;601(7893):366–73.
42. Cha S-H, Hong J, McGuffie M, Yeom B, VanEpps JS, Kotov NA. Shape-dependent biomimetic inhibition of enzyme by nanoparticles and their antibacterial activity. ACS Nano. 2015;9(9):9097–105.
43. Kadiyala U, Turali-Emre ES, Bahng JH, Kotov NA, VanEpps JS. Unexpected insights into antibacterial activity of zinc oxide nanoparticles against methicillin resistant *Staphylococcus aureus* (MRSA). Nanoscale. 2018;10(10):4927–39.
44. Patra JK, Das G, Fraceto LF, Campos EVR, del Pilar Rodriguez-Torres M, Acosta-Torres LS, Diaz-Torres LA, Grillo R, Swamy MK, Sharma S, et al. Nano based drug delivery systems: recent developments and future prospects. J Nanobiotechnol. 2018;16(1):71.
45. Duncan R. Polymer conjugates as anticancer nanomedicines. Nat Rev Cancer. 2006;6(9):688–701.

46. Bouffard E, El Cheikh K, Gallud A, Da Silva A, Maynadier M, Basile I, Gary-Bobo M, Morere A, Garcia M. Why anticancer nanomedicine needs sugars? Curr Med Chem. 2015;22(26):3014–24.
47. Torrice M. Does nanomedicine have a delivery problem?. ACS Publications;2016.
48. Zamboni WC, Torchilin V, Patri AK, Hrkach J, Stern S, Lee R, Nel A, Panaro NJ, Grodzinski P. Best practices in cancer nanotechnology: perspective from NCI nanotechnology alliance. Clin Cancer Res. 2012;18(12):3229–41.
49. Fukuhara N, Kawabata T. HOMCOS: a server to predict interacting protein pairs and interacting sites by homology modeling of complex structures. Nucleic Acids Res. 2008;36:185–9. https://doi.org/10.1093/nar/gkn218.
50. Ghoorah AW, Devignes M-D, Smaïl-Tabbone M, Ritchie DW. Spatial clustering of protein binding sites for template based protein docking. Bioinformatics. 2011;27(20):2820–7. https://doi.org/10.1093/bioinformatics/btr493.
51. Ohue M, Matsuzaki Y, Uchikoga N, Ishida T, Akiyama Y. MEGADOCK: an all-to-all protein–protein interaction prediction system using tertiary structure data. Protein Peptide Lett. 2013;21(8):766–78. https://doi.org/10.2174/09298665113209990050.
52. Szilagyi A, Zhang Y. Template-based structure modeling of protein–protein interactions. Curr Opin Struct Biol. 2014;24:10–23. https://doi.org/10.1016/j.sbi.2013.11.005.
53. Zhang QC, Petrey D, Deng L, Qiang L, Shi Y, Thu CA, Bisikirska B, Lefebvre C, Accili D, Hunter T, Maniatis T, Califano A, Honig B. Structure-based prediction of protein–protein interactions on a genome-wide scale. Nature. 2012;490(7421):556–60. https://doi.org/10.1038/nature11503.
54. Northey TC, Barešić A, Martin ACR. IntPred: a structure-based predictor of protein–protein interaction sites. Bioinformatics. 2018;34(2):223–9. https://doi.org/10.1093/bioinformatics/btx585.
55. Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. Neural message passing for quantum chemistry. In: Proceedings of the 34th international conference on machine learning-volume 70;2017. JMLR. org. p. 1263–72
56. Baranwal M, Magner A, Elvati P, Saldinger J, Violi A, Hero AO. A deep learning architecture for metabolic pathway prediction. Bioinformatics. 2019.
57. Jiang W, Qu Z-B, Kumar P, Vecchio D, Wang Y, Ma Y, Bahng JH, Bernardino K, Gomes WR, Colombari FM, et al. Emergence of complexity in hierarchically organized chiral particles. Science. 2020.
58. Zhu H, Du X, Yao Y. Convsppis: Identifying protein–protein interaction sites by an ensemble convolutional neural network with feature graph. Curr Bioinform. 2020;15(4):368–78.
59. Lehne B, Schlitt T. Protein–protein interaction databases: keeping up with growing interactomes. Hum Genomics. 2009;3(3):291.
60. Orchard S, Ammari M, Aranda B, Breuza L, Briganti L, Broackes-Carter F, Campbell NH, Chavali G, Chen C, Del-Toro N, et al. The MIntAct project-intact as a common curation platform for 11 molecular interaction databases. Nucleic Acids Res. 2014;42(D1):358–63.
61. Szklarczyk D, Gable AL, Lyon D, Junge A, Wyder S, Huerta-Cepas J, Simonovic M, Doncheva NT, Morris JH, Bork P, et al. String v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. Nucleic Acids Res. 2019;47(D1):607–13.
62. Trabuco LG, Betts MJ, Russell RB. Negative protein–protein interaction datasets derived from large-scale two-hybrid experiments. Methods. 2012;58(4):343–8.
63. Bateman A. Uniprot: a universal hub of protein knowledge. In: Protein science, vol. 28. Wiley 111 River St, Hoboken 07030-5774, NJ USA; 2019. p. 32.
64. Burley SK, Berman HM, Bhikadiya C, Bi C, Chen L, Di Costanzo L, Christie C, Dalenberg K, Duarte JM, Dutta S, et al. RCSB protein data bank: biological macromolecular structures enabling research and education in fundamental biology, biomedicine, biotechnology and energy. Nucleic Acids Res. 2019;47(D1):464–74.
65. Borgwardt KM, Ong CS, Schönauer S, Vishwanathan S, Smola AJ, Kriegel H-P. Protein function prediction via graph kernels. Bioinformatics. 2005;21(1):47–56.
66. Ralaivola L, Swamidass SJ, Saigo H, Baldi P. Graph kernels for chemical informatics. Neural Netw. 2005;18(8):1093–110.
67. Pires DE, Ascher DB, Blundell TL. MCSM: predicting the effects of mutations in proteins using graph-based signatures. Bioinformatics. 2014;30(3):335–42.
68. Tsubaki M, Tomii K, Sese J. Compound-protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. Bioinformatics. 2018;35(2):309–18.
69. Xu K, Ba J, Kiros R, Cho K, Courville A, Salakhudinov R, Zemel R, Bengio Y. Show, attend and tell: neural image caption generation with visual attention. In: International conference on machine learning. 2015. p. 2048–57.
70. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, et al. Scikit-learn: machine learning in python. J Mach Learn Res. 2011;12:2825–30.
71. Ben-Hur A, Ong CS, Sonnenburg S, Schölkopf B, Rätsch G. Support vector machines and kernels for computational biology. PLoS Comput Biol. 2008;4(10):1000173.
72. Du X, Sun S, Hu C, Yao Y, Yan Y, Zhang Y. DeepPPI: Boosting prediction of protein–protein interactions with deep neural networks. GitHub; 2017.
73. Yao Y, Du X, Diao Y, Zhu H. An integration of deep learning with feature embedding for protein–protein interaction prediction. GitHub; 2019.
74. Ketkar N. Introduction to Keras. Apress, Berkeley, CA; 2017. p. 97–111. https://doi.org/10.1007/978-1-4842-2766-4_7
75. Ketkar N. Introduction to PyTorch. Apress, Berkeley, CA; 2017. p. 195–208. https://doi.org/10.1007/978-1-4842-2766-4_12
76. Kingma DP, Ba J. Adam: A method for stochastic optimization 2014. arXiv:1412.6980
77. Dick K, Green JR. Reciprocal perspective for improved protein–protein interaction prediction. Sci Rep. 2018;8(1):1–12.
78. Brown G, Bun M, Feldman V, Smith A, Talwar K. When is memorization of irrelevant training data necessary for high-accuracy learning? In: Proceedings of the 53rd annual ACM SIGACT symposium on theory of computing. 2021. p. 123–32.
79. Chu M, Zhou M, Jiang C, Chen X, Guo L, Zhang M, Chu Z, Wang Y. Staphylococcus aureus phenol-soluble modulins α1-α3 act as novel toll-like receptor (TLR) 4 antagonists to inhibit hmgb1/tlr4/nf-κb signaling pathway. Front Immunol. 2018;9:862.
80. Ponnuraj K, Bowden MG, Davis S, Gurusiddappa S, Moore D, Choe D, Xu Y, Hook M, Narayana SV. A "dock, lock, and latch" structural model for a staphylococcal adhesin binding to fibrinogen. Cell. 2003;115(2):217–28.

81. Mehlin C, Headley CM, Klebanoff SJ. An inflammatory polypeptide complex from staphylococcus epidermidis: isolation and characterization. J Exp Med. 1999;189(6):907–18.
82. Tayeb-Fligelman E, Tabachnikov O, Moshe A, Goldshmidt-Tran O, Sawaya MR, Coquelle N, Colletier J-P, Landau M. The cytotoxic staphylococcus aureus PSMα3 reveals a cross-α amyloid-like fibril. Science. 2017;355(6327):831–3.
83. Wang Y, Weng H, Song JF, Deng YH, Li S, Liu HB. Activation of the HMGB1-TLR4-NF-κB pathway may occur in patients with atopic eczema. Mol Med Rep. 2017;16(3):2714–20.
84. Patti JM, Höök M. Microbial adhesins recognizing extracellular matrix macromolecules. Curr Opin Cell Biol. 1994;6(5):752–8.
85. Fu X, Wang Y, Song X, Shi X, Shao H, Liu Y, Zhang M, Chang Z. Subunit interactions as mediated by "non-interface" residues in living cells for multiple homo-oligomeric proteins. Biochem Biophys Res Commun. 2019;512(1):100–5. https://doi.org/10.1016/j.bbrc.2019.03.004.
86. David A, Sternberg MJ. The contribution of missense mutations in core and rim residues of protein–protein interfaces to human disease. J Mol Biol. 2015;427(17):2886–98.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.