BMC Bioinformatics

RESEARCH

Open Access

Protein–protein interaction site prediction by model ensembling with hybrid feature and self-attention

Hanhan Cong[1,2], Hong Liu[1,2*], Yi Cao[3,4], Cheng Liang[1] and Yuehui Chen[3,4]

*Correspondence:
lhsdcn@126.com

[1] School of Information Science and Engineering, Shandong Normal University, Jinan, China
[2] Shandong Provincial Key Laboratory for Novel Distributed Computer Software Technology, Jinan, China
[3] School of Information Science and Engineering, University of Jinan, Jinan, China
[4] Shandong Provincial Key Laboratory of Network Based Intelligent Computing, Jinan, China

Abstract

Background: Protein–protein interactions (PPIs) are crucial in various biological functions and cellular processes. Thus, many computational approaches have been proposed to predict PPI sites. Although significant progress has been made, these methods still have limitations in encoding the characteristics of each amino acid in sequences. Many feature extraction methods rely on the sliding window technique, which simply merges all the features of residues into a vector. The importance of some key residues may be weakened in the feature vector, leading to poor performance.

Results: We propose a novel sequence-based method for PPI sites prediction. The new network model, PPINet, contains multiple feature processing paths. For a residue, the PPINet extracts the features of the targeted residue and its context separately. These two types of features are processed by two paths in the network and combined to form a protein representation, where the two types of features are of relatively equal importance. The model ensembling technique is applied to make use of more features. The base models are trained with different features and then ensembled via stacking. In addition, a data balancing strategy is presented, by which our model can get significant improvement on highly unbalanced data.

Conclusion: The proposed method is evaluated on a fused dataset constructed from Dset186, Dset_72, and PDBset_164, as well as the public Dset_448 dataset. Compared with current state-of-the-art methods, the performance of our method is better than the others. In the most important metrics, such as AUPRC and recall, it surpasses the second-best programmer on the latter dataset by 6.9% and 4.7%, respectively. We also demonstrated that the improvement is essentially due to using the ensemble model, especially, the hybrid feature. We share our code for reproducibility and future research at https://github.com/CandiceCong/StackingPPINet.

Keywords: Protein–protein interaction, Hybrid feature, Self-attention, Integration framework

Background

Protein–protein interactions (PPIs) play a crucial role in various biological functions and cellular processes [1], such as signal transduction, immunological recognition, metabolism [2] etc. During PPIs, some interfaces are formed at particular protein residues, called protein–protein interaction sites [3]. Therefore, identifying those sites are essential to reveal the key mechanisms of PPIs and beneficial to modern drug design [4, 5]. However, via experiments, PPI sites identification requires high-end devices and accurate manipulations, being time-consuming and expensive. As an economic and efficient alternative, computational methods [6] have been widely applied. In particular, data-driven methods can provide competitive results by leveraging machine learning and modern deep learning techniques [7–10]. Existing computational approaches can be roughly divided into partner-independent prediction [11] and partner-specific prediction [12]. In addition, according to the feature information, partner-independent prediction can be further divided into structure-based methods [13] and sequence-based methods [14]. Structure-based methods usually need structural details [15], while the structural information for many proteins is currently unavailable in the dataset. With the rapid development of high-throughput sequencing techniques, a growing number of protein sequences can be obtained, which attracts more attention for sequence-based methods [16].

Since the functions of the residues are determined by its physiochemical properties and context [17–19], residues are usually represented by these properties, e.g., accessible surface area [20], protein sequence composition, hydrophilic and hydrophobic index [21]. In addition, evolutionary information [22] and secondary structure information [23] are often incorporated as supplements. To model the local context, sliding window-based methods [24] are widely applied. However, the features of the residues in the window are typically treated equally, which is obviously inaccurate and harms the precise PPI site prediction [25]. Hitherto, many machine learning methods have been proposed to deal with this prediction task, including neural networks (NNs) [26], support vector machines (SVMs) [27], random forests (RF) [28], etc. ISIS [29] is a neural network predictor, which is trained on sequences profiles and structural features predicted from the sequences. SPPIDER [30] employs an SVM, neural network and linear discriminant analysis based on 19 selected features from the sequences. SPRINGS [31] uses mean cumulative hydrophobicity, relative solvent accessibility, and structural features to represent the targeted residue site, and the algorithm uses neural networks for classifier construction. DeepPPISP [32] is an end-to-end deep learning framework that combines local contextual and global sequence features to fulfill the prediction task. Although considerable progress has been achieved, the predictive performance of these methods still needs to be improved [33].

As a matter of fact, most residues in proteins are not PPI sites and thus making the data highly imbalanced [34]. The cascade random forests algorithm (CRF) [35] is first proposed to deal with the problem. It connects multiple random forests in a cascade-like manner, each of which is trained with a balanced training subset that includes all minority samples and a subset of majority samples. However, sampling of training data based on residues level might destroy the completeness of a sequence. SSWRF [36] combines an ensemble of SVMs and sample-weighted random forests to solve the imbalance

issue and achieves competing performance. SLSTM utilizes a simplified long short-term memory [37] network to improve the precision of the imbalanced PPI sites. It builds a set of protein sequences, instead of single residues, to retain the entire sequential completeness of each protein. The balancing methods either increase the samples of the minority class or reduces the samples of the majority class, which partly change the data distribution.

In this paper, we proposed a novel sequence-based method for PPI sites prediction. The new network model, namely, PPINet, contains multiple feature processing paths. For a residue, the PPINet extracts the features of the targeted residue and its context separately. These two types of features are processed by two paths and combined to form a protein representation, where the two types of features are of relatively equal importance. The individual PPINets are further ensembled via stacking, by which multiple types of features can be merged. To get high quality hybrid features, the dimensions of the 2 types of the features are adjusted to be equal. Therefore, the bias caused by feature dimensionality can be eliminated during feature fusion. Moreover, a novel data balancing strategy is presented. The majority class of samples (non-interaction sites) are divided into multiple sub-datasets. Each sub-dataset is merged with the entire minority class (interaction sites) to form a balanced dataset, which is used for training. In this way, the consistency of data distributions can be maintained.

Based on the above novelty, the contributions of this paper are as follows:

(1) A hybrid feature representation method is proposed to avoid the drawbacks of the traditional sliding window-based methods. The single targeted residue feature and the context feature based sliding window are extracted. They are processed by 2 paths in the PPINet and combined to form a hybrid feature of a protein. This idea is also extended via stacking, where multiple types of features are merged to form a full representation of a protein.
(2) A new feature fusion method is proposed, where the feature importance is balanced. In each PPINet, 2 feature vectors are concatenated to form a hybrid feature of a protein. Before concatenation, the dimensions of them are adjusted to be equal so that they can be exploited equally by the model. Therefore, the bias caused by feature dimensionality can be eliminated.

Methods

This section describes our proposed ensemble framework (StackingPPINet) for PPI prediction. The architecture of the proposed StackingPPINet is shown in Fig. 1, which fundamentally consists of a group of base classifiers, named PPINets, and a stacking module for ensembling. A PPINet is an independent classifier which predicts whether the targeted amino acid in an input sequence segment is a PPI site. It further contains a feature forming module (FFMod), a feature aggregation network (FANet) and a predictor (PPIPred). The FFMod extracts various low-level features from the input sequence by traditional feature extraction methods. The extracted low-level features are then aggregated into a highly abstracted feature vector with fixed dimension by a FANet. Based on the aggregated feature, decisions are made by the predictor, which is a deep neural

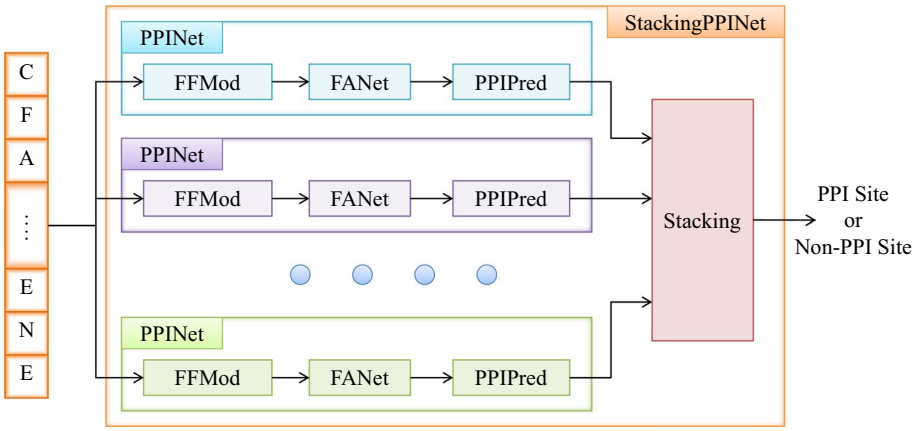

Fig. 1 The architecture of StackingPPINet. StackingPPINet contains multiple base classifiers, named PPINets. Each PPINet consists of a feature forming module (FFMod), a feature aggregation network (FANet) and a prediction module (PPIPred). Multiple PPINets are trained independently and ensembled via Stacking

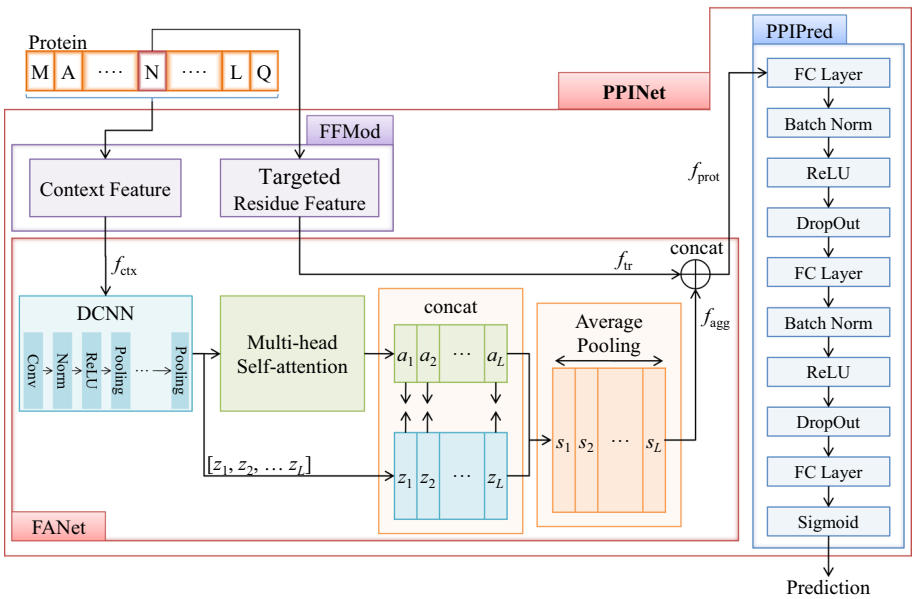

Fig. 2 The architecture of PPINet. The feature forming module (FFMod) extracts low level features from the input protein sequence. As an example, K-PseAA and PhyChem are applied in this showcase to extract the targeted residue feature and the context feature respectively. These features are then processed by a feature aggregation network (FANet). Based on the aggregated features, the predictor (PPIPred) performs classification

network with a binary output. In StackingPPINet architecture, multiple PPINets are first trained independently and then ensembled to enhance the performance and robustness.

The base classifier for protein–protein interaction site prediction

The base classifiers in the StackingPPINet are PPINets, whose architecture is illustrated in Fig. 2. The FFMod extracts various low-level features from the input protein sequence by traditional feature extraction methods. Specifically, f_{tr} is the targeted residue feature for the target residue, while f_{ctx} is the context feature, which is a series of feature vectors

extracted by sliding window-based method. FANet is responsible for aggregating the context feature f_{ctx} into a vector f_{agg} and generating a full protein segment representation, namely f_{prot}, by concatenating f_{tr} and f_{agg}. Finally, predictions are made by PPINet, which is a deep neural network. Details of those modules are demonstrated in the following subsections.

The feature forming module

In many existing works, the entire input sequence is converted into a fixed-dimensional feature vector, or is converted into a vector sequence, where the features are extracted separately from each residue. The features of the predicting residue are treated equally to contextual residues. When the context is extended for including more information, the importance of the targeted residue will be weakened, unintendedly harming the model performance.

To address this issue, FFMod extracts the targeted residue feature f_{tr} and the context feature f_{ctx} separately for a residue. For the targeted residue features, FFMod firstly extracts single features and combine them. In this paper, we use six single features, they are one-hot vector [38], position-specific scoring matrix (PSSM) [39], entropy density (Den) [40], physicochemical properties (PhyChem) [41], hydrophilicity and hydrophobicity index (HyIn) [42], and the pseudo amino acid [43] based on K-nearest neighbors (K-PseAA). And the single features are then concatenated in pairs to form combination features for the targeted residue. Finally, there are three combination features for a targeted residue. Table 1 shows the details of these features. For the context feature, it is a connection of multiple residue features in the sliding window. Since it bases sliding window, zero padding is applied for the residues at the ends of the sequence. Although sliding window-based methods can model regional features to some extent, their feature aggregations are restricted by the window size and their simple aggregation patterns. Thus, FFMod only produces low-level features of the input protein sequence, which are not sufficient for PPI sites prediction. The obtained context feature f_{ctx} will be further processed by FANet. Since those 2 types of features are provided separately, FANet can handle them respectively and balance their relative importance, which will be introduced in the next subsection.

The feature aggregation network

The context feature f_{ctx} provided by FFMod is a vector sequence. If the targeted residue feature f_{tr} is directly concatenated with f_{ctx} to form a full representation of the input protein, the dimension of the context feature is overwhelming, preventing the classifier to exploit the target residue feature. The goal of the FANet is to generate an aggregated feature vector f_{agg} from f_{ctx}, whose dimension is comparable to the dimension of f_{tr}. Correspondingly, FANet contains 2 paths for the targeted residue feature and the context feature respectively, as shown in Fig. 2. The main path performs the feature aggregation for the context feature f_{ctx}. The input vector sequence is first processed by a deep convolutional neural network (DCNN) [44] block, consisting of convolution, ReLU and max pooling operations. In this phase, 1D convolution [45] is applied with zero padding so that the output sequence maintains the same length as the input. In this way, local invariant patterns can be captured by this

Table 1 The feature extraction methods used in this paper

Feature	Abbreviation	Description
One-hot vector	Seq	It composed by 20 types of different amino acids and a 20D one-hot vector is used to encode it
Position-specific scoring matrix	PSSM	It represents the probabilities of 20 amino acids occurring at each position, and the PSI-BLAST algorithm is used to generate it, i.e., we search against the NCBI's non-redundant sequence database with three iterations and an E-value threshold 0.001
Entropy density	Den	It represents the composition information of the protein sequence and obtained by calculating the information entropy of 20 amino acid residues
Physicochemical properties	PhyChem	It represents the physical and chemical attributes of different amino acid residues and obtained by multivariate statistical analysis of 188 natural amino acid properties
Hydrophilicity and hydrophobicity index	HyIn	A larger hydropathic index means that the residue is more hydrophilic. Conversely, the residues will have higher hydrophobic properties. The hydrophobicity index is the opposite
Pseudo amino acid based on K-nearest neighbors	K-PseAA	It is a new feature combining K-nearest neighbors with the PseAA proposed in this paper. A subsequence is formed by combining the targeted amino acid residue with the residues that are not more than K before and after it. The length of the subsequence is 2K + 1. Then we calculate the PseAA of this subsequence as the K-PseAA feature of the targeted amino acid residue

module. Then the output of the DCNN block is further processed by a multi-head self-attention module, which assigns different weights to the features. For each element z_i, a query q_i, a key k_i and a value v_i are generated by the weight matrices W_Q, W_K and W_V as follows:

$$\begin{aligned} &q_i = W_Q z_i, k_i = W_K z_i, v_i = W_V z_i, \\ &W_Q, W_K, W_V \in \mathrm{R}^{d_m \times D}, q_i, k_i, v_i \in \mathrm{R}^{d_m} \end{aligned} \tag{1}$$

where d_m is the feature dimension of each head, and D refers to the total number of convolutional filters in DCNN block. By matrices, Eq. (1) can be rewritten as:

$$\begin{aligned} &Q = W_Q Z, K = W_K Z, V = W_V Z, \\ &Q, K, V \in \mathrm{R}^{d_m \times L} \end{aligned} \tag{2}$$

To calculate the attention weights, an energy score matrix E is calculated with a Mask operation:

$$E = Mask\left(\frac{Q \times K^T}{\sqrt{d_m}}\right) \tag{3}$$

where the correlation matrix $Q \times K^T$ is scaled by $\sqrt{d_m}$ [46]. The Mask operation adds a large penalty to each position in the padding regions, which weakens the attention to those regions. After that, the weights are obtained by a softmax function as follows:

$$w_{i,j} = \frac{exp(e_{i,j})}{\sum_{i=1}^{L} exp(e_{i,j})}, \quad 1 \le i,j \le L \tag{4}$$

where L is the sequence length produced by DCNN block.

Then, the feature of this head at position j is a weighted summation defined as:

$$h_j = \sum_{i=1}^{L} w_{i,j} v_i, \quad 1 \le j \le L \tag{5}$$

$$H_h = \left[h_1, \ldots, h_j, \ldots, h_L\right] \tag{6}$$

By concatenation, the multi-head features are obtained by:

$$\mathbf{A} = \left[a_1, \ldots, a_j, \ldots, a_L\right] = \text{concat}(H_h), \quad 1 \le h \le d_H \tag{7}$$

where H_h is the output feature of a head, d_H is the number of heads. The obtained sequential features z_i and a_i, $1 \le i \le L$, are concatenated in an elementwise manner as Eq. (8):

$$\mathbf{S} = \left[s_1, \ldots, s_j, \ldots, s_L\right], s_i = \text{concat}(z_i, a_i), \quad 1 \le i \le L \tag{8}$$

By concatenation, the features from 2 levels of abstraction can be maintained. Next, an average pooling is adopted across all the elements in **S**, aggregating all features into an information-dense vector as the abstraction of the input sequence.

$$f_{\text{agg}} = \text{AveragePooling}(\mathbf{S}) \tag{9}$$

The protein feature f_{prot} is the concatenation of f_{agg} and f_{tr}, which passes through another path without any transformation. Following traditional design patterns, the input f_{tr} should be transformed by several fully connected layers. However, f_{tr} will be processed by the fully connected layers in the following PPIPred module. It is not necessary to add extra fully connected layers in this path to save some parameters. Similarly, there is no need to add fully connected layers to adjust the dimension of f_{ctx}. Instead, the numbers of convolution kernels and attention heads are carefully controlled so that the dimensions of f_{tr} and f_{agg} are comparable.

The predictor of protein–protein interaction sites

The PPIPred module consists of 3 fully connected layers (FC Layers) with ReLU activation as shown in Fig. 2. To smooth the training, batch normalization is inserted between adjacent fully connected layers. Similarly, DropOut is applied to enhance the generalization. The prediction is produced by a sigmoid activation.

The stacking of multiple base classifiers

As a matter of fact, model performance heavily relies on features. One can conduct a series of experiments to find the optimal feature combinations and train one PPINet as the predictor. However, the results could be misleading due to overfitting when those experiments are based on limited data. As a better alternative, multiple PPINets are trained and then ensembled via stacking [47] in this paper. The ensembled model, called

StackingPPINet, could be more robust thanks to the model diversity. To obtain diverse individual PPINets, each PPINet is trained independently using different data and feature combinations.

Suppose there are K combinations features, each combination corresponding a base classifier which is employed in StackingPPINet. The parameters K is three in this paper. With different feature combinations, multiple PPINets can be trained independently. Then, these diverse models are ensembled via stacking. The final prediction is made by a decision rule in the stacking module as shown in Fig. 2.

Benchmark datasets

In the process of model hyperparameter adjustment, three benchmark datasets, i.e., Dset_186, Dset_72 [48] and PDBset_164 datasets [31], are fused as a dataset, called Dset_186_72_PDB164 in this paper. To maintain consistency with other model training data, we remove two protein sequences as they do not have the definition of secondary structure of proteins (DSSP) file, same as the datasets in [32]. In the fact, we do not use the DSSP feature. There are 422 protein sequences ranging from 39 to 2000 amino acids in the fused dataset, and 61.85% of them contain less than 200 amino acids. An amino acid is defined as a protein–protein interaction (PPI) site if its absolute solvent accessibility is < 1 Å^2, otherwise, it is a non-PPI site. There are 13,536 interaction sites and 74,504 non-interaction sites. Table 2 shows the statistics of those datasets. Dset_186_72_PDB164 is divided into a training set, a validation set, and a test set according to the ratio of 3:1:1, respectively. The divided process complies with two principles, they are random selection, and sites of the same sequence are in the same sub-dataset.

In the comparison with other methods, we first compare the trained model on Dset_186_72_PDB164 with the performance in [32]. The paper uses the fused dataset for model training. And then we evaluate our proposed method with the performance in [17]. We use a large dataset [49] as this paper to train our model, and then do the same test on Dset_448 [50]. The raw data of Dset_448 was from the BioLip database [51]. The statistics of sites in the two datasets show on Table 2.

It is well acknowledged that similar sequences between training and testing datasets negatively affect the generalization of the evaluated performance of a machine learning model. Dset_186 was built based on a PDB collection [52] to which a six-step filtering process was applied to refine the data, including similarities elimination. Dset_72

Table 2 The statistics of all sites in Benchmark datasets

Dataset	Sequences			Interaction sites		Non-interaction sites	All sites
	Number	Average length	Length ≤ 200 (%)	Number	Proportion (%)		
Dset_186	186	195	65.05	5517	15.23	30,702	36,219
Dset_72	72	252	56.94	1923	10.6	16,217	18,140
PDBset_164	164	205	60.37	6096	18.1	27,585	33,681
Dset_186_72_PDB164	422	209	61.85	13,536	15.37	74,504	88,040
Dset_448	448	260	35.94	15,810	13.57	100,690	116,500
The large dataset	9982	426	28.01	427,687	10.05	3,826,511	4,254,198

was constructed based on the protein–protein benchmark set version 3.0 [53], and any sequences showing ≥ 25% sequence identity over a 90% overlap with any of the sequences in Dset_186, using BLASTClust, were removed. Dset_164 with the same filtering technique as for Dset_186 and Dset_72. The raw data was further processed by removing protein fragments, mapping BioLip sequences to UniProt sequences, and clustering, so no similarities above 25% are shared within Dset_448. The sequences from the large training dataset sharing similarities above 25% were removed, as measured by PSI-CD-hit [54].

Data balancing strategy

Since the data sets for PPI site prediction problem are usually highly unbalanced, traditional oversampling and subsampling methods do not work well. Here, we first construct a series of subsets, where the samples are relatively balanced. Then, we use subsampling to balance all the subsets, which are used for model training. To do so, we first compute the ratio between PPI sites and non-PPI sites, as shown in Eq. (10):

$$M = \frac{N_{r\_n}}{N_{r\_p}} \tag{10}$$

where N_{r_n} and N_{r_p} are non-PPI sites and PPI sites in the dataset. Usually, non-PPI sites are far more than PPI sites. Hence, $M > 1$. Then, we divide non-PPI sites into M parts. Each part of the non-PPI sites is combined with all PPI sites to form a subset, where the ratio of non-PPI sites to PPI sites is less than 2. The constructed M subsets are fed to the PPINets for training. During training, each subset is further balanced by subsampling. In this way, when all non-PPI sites are fed to the networks, PPI sites have been learned M times. To some extent, PPI sites are oversampled.

Implementation details

Our model is implemented by PyTorch (http://pytorch.org/). The loss function for StackingPPINet is mean square error (MSE), while the loss for training the individual PPINet is the cross-entropy loss, defined as follows:

$$Loss = -\frac{1}{n}\sum \left[y \log\left(y_{pred}\right) + \left(1-y\right) \log\left(1-y_{pred}\right)\right] \tag{11}$$

where n is the number of all training samples, y is the label and y_{pred} is the model prediction.

The program is written in Python 3.7.4 with PyTorch 1.8.1 + cu101 as the back end. All features are computed from protein sequences only. According to the methods proposed in [38–43], we have implemented feature extraction functions used in the paper in Python, which have been published on GitHub. The parameters of the feature extraction methods are given in Table 3. The structure and parameters of the model are shown in Table 4. The length of the sliding window for context features is discussed in the experimental section, where the window length of 8, 16, 32, and 64 are considered. The threshold is set to 0.5 for the final decision.

We trained our model on the training set with the Adam optimizer [55]. To avoid overfitting, DropOut is applied after the first and the second fully connected layer

Table 3 The parameters of the feature extraction method in each FFMod

Component	FFMod	Parameter	Value
Seq	0	Dimension	20
Den	0	Dimension	20
PhyChem	1	Dimension	21
HyIn	1	Dimension	2
PSSM	2	Dimension	20
K-PseAA	2	Max level correlation factor	10
		Dimension	30

Table 4 The modules and parameters of the model in the experiment

Component	FFMod	Parameter	Value
Convolutional layers	0,1,2	Kernel size (1-Dimensional)	5,5,5
		Number of Kernels	8,8,8
		Strides	1,1,1
		Activation function	ReLU
Pooling layers	0,1,2	Size (1-Dimensional)	3,3,3
		Strides	1,1,1
Self-attention	0,1,2	Heads	4
		Attention-dimension	16
Fully connected layer	0,1,2	Neurons	1024
		Neurons	256
		Neurons	1
		Activation function	ReLU,ReLU,Sigmoid
		DropOut rate	0.5

Table 5 The training parameters in experiment

Parameter	Value
Optimizer	Adam with default parameters
Learning rate	0.001
Batch size	64
Max epoch	50

with the rate of 0.5. The training stops when the average loss of the last 3 epochs continues to increase for 5 epochs or the maximum epochs of 50 is reached. Meanwhile, the independent validation set is also used to tune hyper parameters and perform model selection, such as choosing different ensemble methods and convolutional neural network architectures. Finally, the model is tested on an independent test set. The training and testing are conducted on a workstation with a GTX 1660Ti graphics card and 32 GB RAM. The training parameters are listed in Table 5.

Results

Evaluation metrics

We assume the PPI sites are the positive samples and the non-PPI sites are the negative samples. To evaluate the performance, we use five evaluation metrics. They are accuracy (ACC), precision (Pre), recall (Rec), F1 scores (F1) and Matthew's correlation coefficient (MCC). The calculations of these measurements are:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

$$Pre = \frac{TP}{TP + FP} \tag{13}$$

$$Rec = \frac{TP}{TP + FN} \tag{14}$$

$$F1 = \frac{2 * Pre * Rec}{Pre + Rec} \tag{15}$$

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{16}$$

where TP, TN, FP and FN represent true positives, true negatives, false positives, and false negatives, respectively. Area under the ROC curve (AUROC) and area under the precision-recall curve (AUPRC) are also used for evaluations [56].

Performance comparison of StackingPPINet and other PPI predictors

To evaluate the performance of the proposed method, we have compared it with six state-of-the-art machine-learning-based methods on the Dset_186_72_PDB164. They are PSIVER [48], SPPIDER [30], SPRINGS [31], ISIS [29], RF_PPI [28] and DeepPPISP [32]. PSIVER uses the PSSM and solvent accessibility within a sliding window to represent the feature of the targeted residue site, and it employs a naive Bayes classifier for prediction. RF_PPI uses a variety of feature representations and employs a random forest classifier for PPI sites prediction. The other four model are described in Section Background. Among these methods, ISIS, SPRINGS and DeepPPISP are neural network models, while SPPIDER uses a SVM classifier.

In the experiment, we use the same datasets to train our model as the other six methods. At last, we use the same set as the other six methods for testing. Table 6 shows the predictive performance of different methods. It can be seen from the experiment results that StackingPPINet achieves better performance than the other algorithms in terms of all evaluation metrics except ACC. With respect to Rec, our method obtains the highest value of 0.683, which is 0.106 over the second-best method. For Pre, F1 and MCC, the results of our method also demonstrate significant advantages over those of the completing alternatives. In summary, these results clearly show the superiority of our method in reliably predicting the PPI sites. As the

Table 6 Predictive performance of different methods on the Dset_186_72_PDB164

Method	ACC	Pre	Rec	F1	MCC	AUROC	AUPRC
PSIVER	0.653	0.253	0.468	0.328	0.138		
SPPIDER	0.622	0.209	0.459	0.287	0.089		
SPRINGS	0.631	0.248	0.598	0.35	0.181		
ISIS	0.694	0.211	0.362	0.267	0.097		
RF_PPI	0.598	0.173	0.512	0.258	0.118		
DeepPPISP	0.655	0.303	0.577	0.397	0.206	0.65	0.68
StackingPPINet	0.597	0.530	0.683	0.582	0.226	0.65	0.77

Table 7 The *p* values of the two-tailed *t*-test for the metrics on the Dset_186_72_PDB164

Method	*p* value of Pre	*p* value of Rec	*p* value of F1	*p* value of MCC
PSIVER	< 0.0001	< 0.0001	< 0.0001	0.1158
SPPIDER	< 0.0001	< 0.0001	< 0.0001	0.0008
SPRINGS	< 0.0001	0.0008	< 0.0001	0.3024
ISIS	< 0.0001	< 0.0001	< 0.0001	0.0016
RF_PPI	< 0.0001	< 0.0001	< 0.0001	0.0136
DeepPPISP	< 0.0001	< 0.0001	< 0.0001	0.0228

DeepPPISP achieves suboptimal performance on the aggregate metrics, we further compare the AUROC and AUPRC of StackingPPINet with DeepPPISP. The AUROC considers the classification of positive and negative samples at the same time. It can be seen that the performance of StackingPPINet and DeepPPISP is basically the same. AUPRC is better suited to evaluate unbalanced data classification. On this metric, the performance of StackingPPINet is clearly better than that of the DeepPPISP. In addition, DeepPPISP uses the secondary structure information of protein sequences. Compared with DeepPPISP, the features used in our model uses are easier to obtain.

Table 7 provides the *p* values of the two-tailed t-test for the metrics on the Dset_186_72_PDB164 data set. From this table, it can be seen that StackingPPINet considerably outperforms other methods in terms of Precision, Recall, and F1. For MCC, StackingPPINet outperforms other methods except for SPRINGS and DeepPPISP. SPRINGS achieves a similar MCC as StackingPPINet, while DeepPPISP obtains significantly better MCC than StackingPPINet. In addition, DeepPPISP performs slightly better than StackingPPINet in terms of AUROC and AUPRC, with the *p* values of 0.0479 and 0.0593, respectively.

To further evaluate the performance of our proposed method, we also compared it with nine state-of-the-art machine-learning-based methods on the Dset_448. They are SCRIBER [50], SSWRF [36], SPRINT [57], CRFPPI [35], LORIS [14], SPRINGS [31], PSIVER [48], SPPIDER [30] and DELPHI [17]. They are also sequence-based methods as sequence information is readily available for most proteins. The evaluation of the other programmers comes from [17]. In the experiment, we use the same datasets to train our model as the other nine methods. Table 8 shows the predictive performance of different methods.

Table 8 Predictive performance of different methods on the Dset_448

Method	Pre	Rec	F1	MCC	AUROC	AUPRC
SPPIDER	0.194	0.202	0.198	0.071	0.517	0.159
SPRINT	0.183	0.183	0.183	0.057	0.570	0.167
PSIVER	0.191	0.191	0.191	0.066	0.581	0.170
SPRINGS	0.228	0.229	0.229	0.111	0.625	0.201
LORIS	0.263	0.264	0.263	0.151	0.656	0.228
CRFPPI	0.264	0.268	0.266	0.154	0.681	0.238
SSWRF	0.286	0.288	0.287	0.178	0.687	0.256
SCRIBER	0.332	0.334	0.333	0.230	0.715	0.287
DELPHI	0.371	0.371	0.371	0.272	0.737	0.337
StackingPPINet	0.360	0.418	0.387	0.129	0.593	0.406

Table 9 Predictive performance of using unbalanced and balanced datasets

Method	ACC	Pre	Rec	F1	MCC	AUROC	AUPRC
Unbalanced datasets	0.795	0.076	0.205	0.111	0.025	0.533	0.189
Balanced datasets	0.549	0.489	0.565	0.512	0.105	0.571	0.554

It can be seen from the experiment results that StackingPPINet achieves the best performance in the most important metrics, such as AUPRC and Rec. It surpasses the second-best programmer by 0.069 and 0.047, respectively. This shows that our algorithm achieves the best results when considering both interaction and non-interaction sites on unbalanced datasets.

Discussion

We introduce an ensemble framework, StackingPPINet, for PPI sites prediction. To demonstrate its performances, we compare it with twelve other PPI sites prediction methods. Based on the design of StackingPPINet and the results of the experiments, we identified five issues worth further discussion. They are the effect of balancing dataset, the stacking ensemble method and its integrated rules, the effectiveness of hybrid feature, the performance on sequences of different lengths. We discuss these issues as follows.

The improvement of using multiple balanced datasets

In the experiment, we compare the predictive performance of the classifiers trained by the unbalanced datasets and the balanced datasets under the same model settings, respectively. We construct the balanced datasets as above described. When training with an unbalanced dataset, the training dataset for each epoch is the entire original dataset. The model structure and parameters of the two experiments are the same. The difference between the two is only whether the datasets using in the training process is processed with the balance strategy proposed in the paper. The stacking adopts the logistic regression to integrate the primary results. The parameters of the classifier model are detailed in Table 4. The length of sliding window for the context feature are 16.

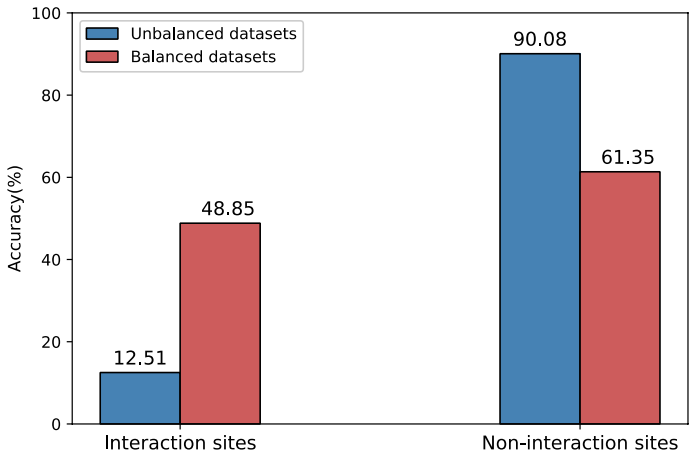

Fig. 3 The accuracy of interaction and non-interaction sites obtained by unbalanced and balanced datasets

Table 10 Predictive performance of using different ensemble methods

Method	ACC	Pre	Rec	F1	MCC	AUROC	AUPRC
Stacking	0.549	0.489	0.565	0.512	0.105	0.571	0.554
Voting	0.531	0.586	0.532	0.557	0.062		
Averaging	0.552	0.616	0.549	0.581	0.104	0.565	0.551

Table 9 shows the performance of our model using unbalanced and balanced datasets. ACC, Pre, Rec, F1, and MCC obtained with the balanced datasets are 0.549, 0.489, 0.565, 0.512, and 0.105, respectively. The results under all evaluation metrics are improved comparing with the results trained with unbalanced datasets except ACC. Especially in F1 and MCC, which reflect the comprehensive performance, the indicators increase from 0.111 to 0.512, and from 0.025 to 0.105, respectively. Since non-interaction sites are far more abundant than interaction sites, the classifier is inclined to the majority category, which simply achieves high ACC and produce deceptive performance. Figure 3 shows the accuracy of interaction and non-interaction sites obtained by the unbalanced and the balanced datasets, respectively. The precision of the non-PPI sites obtained by the unbalanced datasets is relatively high, while the precision of the PPI sites is extremely low. As a matter of fact, it is more important to correctly classify PPI sites than non-PPI sites in practice. Therefore, we use multiple balanced datasets to improve the precision of the PPI sites.

The improvement by stacking

In the proposed method, we use a stacking ensemble method to integrate the prediction of primary classifiers. We here focus on whether the PPI sites prediction could indeed benefit from the stacking method. To this end, we keep the other parts of our model unchanged and replace the ensemble method with either a voting or an averaging mechanism for final prediction. We then compare the prediction results obtained by the three models. Among them, stacking adopts logistic regression as the ensemble rule. The

Table 11 Predictive performance of using and not using stacking

Method	ACC	Precision	Recall	F_value	MCC	AUROC	AUPRC
PPINet 0	0.508	0.508	0.710	0.592	0.013	0.509	0.516
PPINet 1	0.527	0.540	0.425	0.475	0.057	0.549	0.539
PPINet 2	0.543	0.544	0.578	0.560	0.087	0.562	0.551
Stacking	0.549	0.489	0.565	0.512	0.105	0.571	0.554

Table 12 Predictive performance of using different integrated rules

Method	ACC	Pre	Rec	F1	MCC	AUROC	AUPRC
Logistic regression	0.549	0.489	0.565	0.512	0.105	0.571	0.554
Decision tree	0.513	0.476	0.518	0.496	0.027	0.492	0.518
Random forest	0.518	0.482	0.524	0.501	0.038	0.485	0.490
Nearest neighbor	0.516	0.497	0.521	0.507	0.033	0.496	0.527

parameters of the classifier are detailed in Table 4. The length of sliding window for the context feature are 16.

Table 10 shows the performance of the three ensemble methods for predicting PPI sites. The voting method does not end up with a probabilistic calculation, so its AUROC and AUPRC values are not calculated. Obviously, the stacking achieves the best Rec, MCC, AUROC and AUPRC while the averaging mechanism reaches the optimal values on the other three metrics. Although the voting method does not obtain the best results under any evaluation metrics, it retains a relatively stable performance.

In addition, we compared the predictions of individual frames and their integrated with stacking. Table 11 shows the performance results. With stacking, the AUROC and AUPRC values are increased, and the model has stronger generalization ability.

The effects of different integrated rules in stacking

In the stacking ensemble method, different integration rules could also impact the prediction results of PPI sites. In our experiments, we compare the prediction results obtained by four different stacking rules, i.e., logistic regression, decision tree, random forest, and nearest neighbor. The parameters of the classifiers are listed in Table 4. The length of sliding window for the context feature are 16.

Table 12 shows the performance of four stacking rules. The logistic regression achieves the best results on all indicators except Pre. Notably, on MCC, its performance is significantly better than the other alternatives, indicating its overall superiority. Taken together, the comparison results show that the logistic regression could obtain better performance than the other three integrated rules.

The effectiveness of hybrid feature

In this subsection, we exhibit the effectiveness of feature combination. We first compare the results using feature combination with the results using individual feature. Then we investigate the predictive performance under different length of sliding window for the context feature. The parameters of the models are listed in Table 4.

Table 13 The effectiveness of feature combination

Method	ACC	Pre	Rec	F1	MCC	AUROC	AUPRC
Targeted and context feature	0.549	0.5	0.567	0.521	0.105	0.574	0.575
Targeted residue feature	0.544	0.444	0.568	0.488	0.099	0.567	0.568
Context feature	0.52	0.487	0.532	0.499	0.046	0.54	0.538

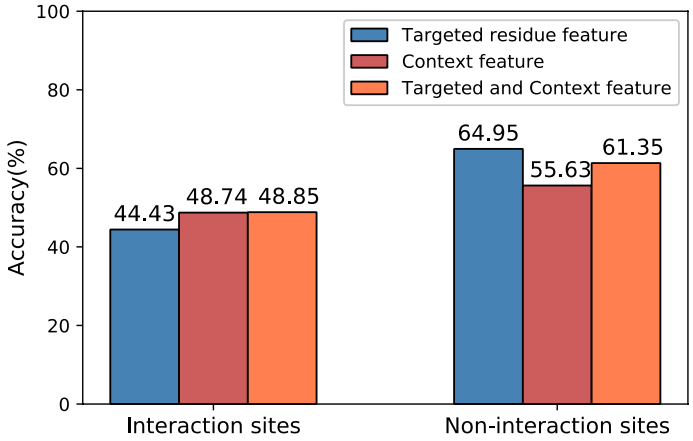

Fig. 4 The accuracy of interaction and non-interaction sites with different features

We compare the predictive performance obtained by different features in experiments. Except the processing module of the context feature and the targeted residue feature, the rest parts are the same. The length of sliding window for the context feature are 16. The detailed results are shown in Table 13. After adding the targeted residue feature, the predictive performance is improved under all evaluation metrics. Specifically, two comprehensive indicators F1 and MCC, increase from 0.499 to 0.521, and 0.046 to 0.105, respectively. This indicates that the targeted residue feature is important to the decision making. Figure 4 shows the improvement of the feature connection for PPI and non-PPI sites. We can see that the addition of the targeted residue feature really improves the accuracy of PPI sites and non-PPI sites.

The effect of sliding window length

The length of sliding window for the context feature is the amino acid range that characterizes the biological properties of the targeted site. In the experiments, we compare the effects of different lengths on the model performance by keeping other model hyper parameters unchanged while varying the lengths of sliding window. It can be seen from the results that the lengths are not as large as possible. If the value is too small, the amino acid residues in the range cannot fully reflect the biological properties. If the value is too large, some amino acid residues in the range may not be related to the interaction of the targeted site. We find that when the length of sliding window for the context feature is 32, our model could reach the best performance. Table 14 shows the effects of different sliding window lengths.

Table 14 The effects of different sliding window lengths

Length	ACC	Pre	Rec	F1	MCC	AUROC	AUPRC
8	0.527	0.461	0.541	0.491	0.060	0.545	0.542
16	0.549	0.489	0.565	0.512	0.105	0.571	0.554
32	0.549	0.500	0.567	0.521	0.105	0.574	0.575
64	0.503	0.471	0.500	0.479	0.005	0.502	0.503

Table 15 Predictive performance on sequences in different lengths

Length	ACC	Pre	Rec	F1	MCC	AUROC	AUPRC
< 100	0.525	0.514	0.777	0.608	0.035	0.508	0.773
100–200	0.548	0.522	0.641	0.562	0.098	0.581	0.643
200–300	0.542	0.507	0.577	0.501	0.121	0.586	0.554
300–400	0.566	0.391	0.439	0.406	0.063	0.549	0.423
> 400	0.562	0.416	0.4	0.395	0.055	0.532	0.397

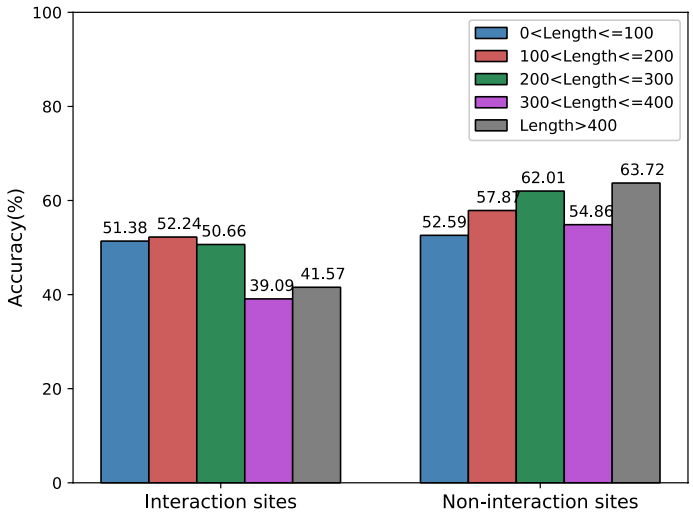

Fig. 5 The accuracy of interaction and non-interaction sites on two categories

The performance on sequences in different lengths

In the experiment, we divide the test set into several subsets and show how performance varies according to the sequence length. As shown in Table 15, the best performance is achieved when the sequence length falls between 100 and 300. When the sequence is too short, the model obtains limited information from the input; when the sequence is too long, the model can be misled by redundant or irrelevant information. Either of those cases may harm the performance, especially for MCC, AUROC, and AUPRC. We also illustrate the accuracy of interaction and non-interaction sites for each subset in Fig. 5. It shows that the accuracy gap rises considerably when the sequence is longer than 200. When the sequence is longer than 400, the accuracy for interaction sites is 22.15% lower than the non-interaction sites. The possible reason could be the data imbalance. There are always enough non-interaction samples (negative samples) for training, while long

interaction samples (positive samples) are relatively limited and more difficult to learn. Thus, for long sequences, our model can get high accuracy for non-interaction samples but considerably lower accuracy for interaction samples.

The effect of multi-head attention

In the PPINet, context features are processed by a CNN block and the multi-head self-attention. Convolutional layers extract features locally, while self-attention aggregates all the sequence information globally. By combining them, it is expected to obtain the global representation of a sequence more efficiently. If the self-attention is removed, one has to add more convolutional layers to extend the reception field to cover the whole sequence. To show the effectiveness of multi-head self-attention, in this experiment, the performance of the model with multi-head attention, the model with single-head attention, and the one without attention are compared. As shown in Table 16, the Base-Model is the one introduced in Table 4 with 4-head self-attention. The SH-Model has the same setup as the Base-Model except that the number of attention heads is only 1. The last NA-Model is constructed by removing the self-attention from the Base-Model and only contains convolutional and fully connected layers. The results show that SH-Model and NA-Model achieve similar ACC, Rec, MCC, AUROC, and AUPRC. SH-Model gains higher Pre and F1. Base-Model outperforms the other two models in all the metrics, indicating the effectiveness of the multi-head attention in global feature aggregation.

Discussion

In the above experiments, only sequence based features are exploited in the proposed model for the sake of fair comparison with considered baseline methods. From the methodology of model ensembling, it can be noticed that the improvement by the proposed ensembling strategy is restricted by the low diversity of based classifiers. To break through such limitation, one feasible way is to introduce multiple types of data, e.g. protein structure features, protein domain features, to train base models. On the one hand, multiple types of information help to construct a full description of a protein; on the other hand, diverse data types require different types of models to process, enhancing the model diversity. Both can bring extra performance gain for model ensembling. The cost of such improvement is the data collection. For a protein, one has to collect multiple types of data to get the prediction, which is not convenient during inference phase. One possible way to further overcome this drawback is to utilize protein language models trained on large sequence data sets. Recent research has reported that accurate protein structure prediction can be achieved by learning from Multiple Sequence Alignment (MSA) data [58, 59] or even pure sequence data [60]. Such models can be used as feature extractors which indirectly introduces protein structure information to base PPINet.

Table 16 Performance comparison of the models with different self-attention setup

Model	ACC	Pre	Rec	F1	MCC	AUROC	AUPRC
Base-model (4-head attention)	0.549	0.489	0.565	0.512	0.105	0.571	0.554
SH-model (Single-head attention)	0.528	0.454	0.540	0.491	0.056	0.539	0.528
NA-model (No attention)	0.522	0.317	0.549	0.375	0.051	0.540	0.543

However, this method has not been extensively studied yet. We decide to leave it to the further work.

Conclusions

In this work, we propose a novel sequence-based method for PPI sites prediction from the motivation of extracting more valuable features. Specifically, we extract the single feature of the targeted amino acid residue and the context feature of its neighbors with different combinations of features to compose the hybrid feature. A deep learning framework combined with convolutional neural networks and multi-head self-attention is employed to process the context feature to control these dimensions. In addition, we present a strategy to balance the interaction sites and non-interaction sites so that the model can ultimately learn the original data distribution. This paper compares the proposed method with the prediction algorithms of twelve existing protein–protein interaction sites. The results show that our method performs well in various indicators, especially on the precision of interaction sites. Though the proposed method is demonstrated to have advantages over other competing methods, it also has some limitations. The first is that the model architecture and the features can be extended. The second is that the optimal parameters of the model are obtained through grid search, which is computationally intensive. Future challenges include exploring more efficient feature expression methods and designing more adaptive network architectures.

Abbreviations

PPIs	Protein–protein interactions
NNs	Neural networks
SVMs	Support vector machines
RF	Random forests
FFMod	Feature forming module
FANet	Feature aggregation network
PSSM	Position-specific scoring matrix
Den	Entropy density
PhyChem	Physicochemical properties
HyIn	Hydrophilicity and hydrophobicity index
K-PseAA	The pseudo amino acid based on K-nearest neighbors
FC Layers	Fully connected layers
DCNN	Deep convolutional neural network
AUROC	Area under the ROC curve
AUPRC	Area under the precision-recall curve

Acknowledgements
We are grateful to those researchers that have made the benchmark datasets available for PPI prediction evaluation.

Author contributions
HC developed the algorithm, did the computation, and wrote the manuscript. YC designed the project, collected the data and revised the manuscript. HL, CL and YC revised the manuscript. All authors read and approved the final manuscript.

Funding
This research was supported by the National Natural Science Foundation of China [61876102]; Shandong Provincial Natural Science Foundation [ZR2021MF036)]; and University Innovation Team Project of Jinan [2019GXRC015].

Availability of data and materials
The datasets supporting the conclusions of this article are available in the Github repository, https://github.com/CandiceCong/StackingPPINet.

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.

Received: 18 December 2022 Accepted: 30 November 2023
Published online: 05 December 2023

References

1. Hu L, Wang X, Huang YA, Hu P, You ZH. A survey on computational models for predicting protein–protein interactions. Brief Bioinform. 2021;22(5):bbab036.
2. Jamasb AR, Day B, Cangea C, Liò P, Blundell TL. Deep learning for protein–protein interaction site prediction. In: Proteomics data analysis. New York, NY: Humana; 2021. p. 263–88.
3. Jordan RA, Yasser EM, Dobbs D, Honavar V. Predicting protein-protein interface residues using local surface structural similarity. BMC Bioinform. 2012;13(1):1–14.
4. Chen M, Ju CJT, Zhou G, Chen X, Zhang T, Chang KW, Wang W, et al. Multifaceted protein–protein interaction prediction based on Siamese residual RCNN. Bioinformatics. 2019;35(14):i305–14.
5. Li X, Li W, Zeng M, Zheng R, Li M. Network-based methods for predicting essential genes or proteins: a survey. Brief Bioinform. 2020;21(2):566–83.
6. Das S, Chakrabarti S. Classification and prediction of protein–protein interaction interface using machine learning algorithm. Sci Rep. 2021;11(1):1–12.
7. Sarkar D, Saha S. Machine-learning techniques for the prediction of protein–protein interactions. J Biosci. 2019;44(4):1–12.
8. Li Y, Wang Z, Li LP, You ZH, Huang WZ, Zhan XK, Wang YB. Robust and accurate prediction of protein–protein interactions by exploiting evolutionary information. Sci Rep. 2021;11(1):1–12.
9. Zhang C, Freddolino PL, Zhang Y. COFACTOR: improved protein function prediction by combining structure, sequence and protein–protein interaction information. Nucleic Acids Res. 2017;45(W1):W291–9.
10. Yang H, Wang M, Liu X, Zhao XM, Li A. PhosIDN: an integrated deep neural network for improving protein phosphorylation site prediction by combining sequence and protein–protein interaction information. Bioinformatics. 2021;37(24):4668–76.
11. Wang X, Yu B, Ma A, Chen C, Liu B, Ma Q. Protein–protein interaction sites prediction by ensemble random forests with synthetic minority oversampling technique. Bioinformatics. 2019;35(14):2395–402.
12. Afsar Minhas FUA, Geiss BJ, Ben-Hur A. PAIRpred: partner-specific prediction of interacting residues from sequence and structure. Proteins Struct Funct Bioinform. 2014;82(7):1142–55.
13. Northey TC, Barešić A, Martin AC. IntPred: a structure-based predictor of protein–protein interaction sites. Bioinformatics. 2018;34(2):223–9.
14. Dhole K, Singh G, Pai PP, Mondal S. Sequence-based prediction of protein–protein interaction sites with L1-logreg classifier. J Theor Biol. 2014;348:47–54.
15. Hou Q, Lensink MF, Heringa J, Feenstra KA. Club-martini: selecting favourable interactions amongst available candidates, a coarse-grained simulation approach to scoring docking decoys. PLoS ONE. 2016;11(5):e0155251.
16. Zhang B, Li J, Quan L, Chen Y, Lü Q. Sequence-based prediction of protein-protein interaction sites by simplified long short-term memory network. Neurocomputing. 2019;357:86–100.
17. Li Y, Golding GB, Ilie L. DELPHI: accurate deep ensemble model for protein interaction sites prediction. Bioinformatics. 2021;37(7):896–904.
18. Tsubaki M, Tomii K, Sese J. Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. Bioinformatics. 2019;35(2):309–18.
19. Lei Y, Li S, Liu Z, Wan F, Tian T, Li S, Zeng J, et al. A deep-learning framework for multi-level peptide–protein interaction prediction. Nat Commun. 2021;12(1):1–10.
20. Miloserdov O. Classifying amorphous polymers for membrane technology basing on accessible surface area of their conformations. Adv Syst Sci Appl. 2020;20(3):91–104.
21. Jones S, Thornton JM. Prediction of protein-protein interaction sites using patch analysis. J Mol Biol. 1997;272(1):133–43.
22. Singh H, Singh S, Raghava GPS. Peptide secondary structure prediction using evolutionary information. BioRxiv. 2019;558791.
23. Balogh RK, Németh E, Jones NC, Hoffmann SV, Jancsó A, Gyurcsik B. A study on the secondary structure of the metalloregulatory protein CueR: effect of pH, metal ions and DNA. Eur Biophys J. 2021;50(3):491–500.
24. 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.
25. Wang X, Zhang Y, Yu B, Salhi A, Chen R, Wang L, Liu Z. Prediction of protein-protein interaction sites through eXtreme gradient boosting with kernel principal component analysis. Comput Biol Med. 2021;134:104516.
26. Chen H, Zhou HX. Prediction of interface residues in protein–protein complexes by a consensus neural network method: test against NMR data. Proteins Struct Funct Bioinform. 2005;61(1):21–35.
27. Chen P, Wong L, Li J. Detection of outlier residues for improving interface prediction in protein heterocomplexes. IEEE/ACM Trans Comput Biol Bioinform. 2012;9(4):1155–65.
28. Hou Q, De Geest PF, Vranken WF, Heringa J, Feenstra KA. Seeing the trees through the forest: sequence-based homo-and heteromeric protein-protein interaction sites prediction using random forest. Bioinformatics. 2017;33(10):1479–87.

29. Ofran Y, Rost B. ISIS: interaction sites identified from sequence. Bioinformatics. 2007;23(2):e13–6.
30. Porollo A, Meller J. Prediction-based fingerprints of protein–protein interactions. Proteins Struct Funct Bioinform. 2007;66(3):630–45.
31. Singh G, Dhole K. Pai PP, Mondal S. SPRINGS: prediction of protein-protein interaction sites using artificial neural networks (No. e266v2). PeerJ PrePrints. 2014.
32. Zeng M, Zhang F, Wu FX, Li Y, Wang J, Li M. Protein–protein interaction site prediction through combining local and global features with deep neural networks. Bioinformatics. 2020;36(4):1114–20.
33. Lu S, Li Y, Nan X, Zhang S. Attention-based convolutional neural networks for protein-protein interaction site prediction. In: 2021 IEEE international conference on bioinformatics and biomedicine (BIBM). IEEE; 2021. p. 141–144.
34. Xie Z, Deng X, Shu K. Prediction of protein–protein interaction sites using convolutional neural network and improved data sets. Int J Mol Sci. 2020;21(2):467.
35. Wei ZS, Yang JY, Shen HB, Yu DJ. A cascade random forests algorithm for predicting protein-protein interaction sites. IEEE Trans Nanobiosci. 2015;14(7):746–60.
36. Wei ZS, Han K, Yang JY, Shen HB, Yu DJ. Protein–protein interaction sites prediction by ensembling SVM and sample-weighted random forests. Neurocomputing. 2016;193:201–12.
37. Zhang B, Li J, Quan L, et al. Sequence-based prediction of protein-protein interaction sites by simplified long short-term memory network. Neurocomputing. 2019;357:86–100.
38. Al-Shehari T, Alsowail RA. An insider data leakage detection using one-hot encoding, synthetic minority oversampling and machine learning techniques. Entropy. 2021;23(10):1258.
39. Zhang S, Liang Y. Predicting apoptosis protein subcellular localization by integrating auto-cross correlation and PSSM into Chou's PseAAC. J Theor Biol. 2018;457:163–9.
40. Kothawala D, Padmanabhan T. Entropy density of spacetime from the zero point length. Phys Lett B. 2015;748:67–9.
41. Wihodo M, Moraru CI. Physical and chemical methods used to enhance the structure and mechanical properties of protein films: a review. J Food Eng. 2013;114(3):292–302.
42. Abskharon R, Wang F, Wohlkonig A, Ruan J, Soror S, Giachin G, Steyaert J, et al. Structural evidence for the critical role of the prion protein hydrophobic region in forming an infectious prion. PLoS Pathog. 2019;15(12):e1008139.
43. Cong H, Liu H, Chen Y, Cao Y. Self-evoluting framework of deep convolutional neural network for multilocus protein subcellular localization. Med Biol Eng Comput. 2020;58(12):3017–38.
44. Sui X, Zheng Y, Wei B, Bi H, Wu J, Pan X, Zhang S, et al. Choroid segmentation from optical coherence tomography with graph-edge weights learned from deep convolutional neural networks. Neurocomputing. 2017;237:332–41.
45. Mohapatra S, Nayak J, Mishra M, Pati GK, Naik B, Swarnkar T. Wavelet transform and deep convolutional neural network-based smart healthcare system for gastrointestinal disease detection. Interdiscip Sci Comput Life Sci. 2021;13(2):212–28.
46. Vaswani A, Shazeer N, Parmar N. Uszkoreit J, Jones L, Gomez AN, Polosukhin I, et al. Attention is all you need. Adv Neural Inf Process Syst. 2017;30.
47. Kardani N, Zhou A, Nazem M, Shen SL. Improved prediction of slope stability using a hybrid stacking ensemble method based on finite element analysis and field data. J Rock Mech Geotech Eng. 2021;13(1):188–201.
48. Murakami Y, Mizuguchi K. Applying the Naïve Bayes classifier with kernel density estimation to the prediction of protein–protein interaction sites. Bioinformatics. 2010;26(15):1841–8.
49. Zhang J, Ma Z, Kurgan L. Comprehensive review and empirical analysis of hallmarks of DNA-, RNA-and protein-binding residues in protein chains. Brief Bioinform. 2019;20(4):1250–68.
50. Zhang J, Kurgan L. SCRIBER: accurate and partner type-specific prediction of protein-binding residues from proteins sequences. Bioinformatics. 2019;35(14):i343–53.
51. Yang J, Roy A, Zhang Y. BioLiP: a semi-manually curated database for biologically relevant ligand–protein interactions. Nucleic Acids Res. 2012;41(D1):D1096–103.
52. Berman HM, Battistuz T, Bhat TN, et al. The protein data bank. Acta Crystallogr D Biol Crystallogr. 2002;58(6):899–907.
53. Hwang H, Pierce B, Mintseris J, et al. Protein–protein docking benchmark version 3.0. Proteins Struct Funct Bioinform. 2008;73(3):705–9.
54. Fu L, Niu B, Zhu Z, et al. CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics. 2012;28(23):3150–2.
55. Bock S, Goppold J, Weiß M. An improvement of the convergence proof of the ADAM-Optimizer. arXiv preprint arXiv:1804.10587. 2018.
56. Zeng M, Zou B, Wei F, Liu X, Wang L. Effective prediction of three common diseases by combining SMOTE with Tomek links technique for imbalanced medical data. In: 2016 IEEE international conference of online analysis and computing science (ICOACS). IEEE; 2016. p. 225–228
57. Taherzadeh G, Yang Y, Zhang T, et al. Sequence-based prediction of protein–peptide binding sites using support vector machine. J Comput Chem. 2016;37(13):1223–9.
58. Rives A, Meier J, Sercu T, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. Proc National Acad Sci U S A. 2021;118(15):e2016239118.
59. Roshan R, Jason L, Robert V, et al. MSA transformer. In: 38th international conference on machine learning. 2021.
60. Fang X, Wang F, Liu L, et al. A method for multiple-sequence-alignment-free protein structure prediction using a protein language model. Nat Mach Intell. 2023;5:1087–96.

Publisher's Note