frontiers
in Bioengineering and Biotechnology

ORIGINAL RESEARCH
published: 05 May 2020
doi: 10.3389/fbioe.2020.00390

# Protein Interaction Network Reconstruction Through Ensemble Deep Learning With Attention Mechanism

**_Feifei Li[1], Fei Zhu[1,2]*, Xinghong Ling[1] and Quan Liu[1]_**

_[1] School of Computer Science and Technology, Soochow University, Suzhou, China, [2] Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou, China_

Protein interactions play an essential role in studying living systems and life phenomena. A considerable amount of literature has been published on analyzing and predicting protein interactions, such as support vector machine method, homology-based method and similarity-based method, each has its pros and cons. Most existing methods for predicting protein interactions require prior domain knowledge, making it difficult to effectively extract protein features. Single method is dissatisfactory in predicting protein interactions, declaring the need for a comprehensive method that combines the advantages of various methods. On this basis, a deep ensemble learning method called EnAmDNN (Ensemble Deep Neural Networks with Attention Mechanism) is proposed to predict protein interactions which is an appropriate candidate for comprehensive learning, combining multiple models, and considering the advantages of various methods. Particularly, it encode protein sequences by the local descriptor, auto covariance, conjoint triad, pseudo amino acid composition and combine the vector representation of each protein in the protein interaction network. Then it takes advantage of the multi-layer convolutional neural networks to automatically extract protein features and construct an attention mechanism to analyze deep-seated relationships between proteins. We set up four different structures of deep learning models. In the ensemble learning model, second layer data sets are generated with five-fold cross validation from basic learners, then predict the protein interaction network by combining 16 models. Results on five independent PPI data sets demonstrate that EnAmDNN achieves superior prediction performance than other comparing methods.

**Keywords: protein-protein interaction network, protein-protein interaction, ensemble learning, deep learning, attention mechanism, multi-layer convolutional neural network**

OPEN ACCESS

***Edited by:***
_Jiajie Peng,
Northwestern Polytechnical University, China_

***Reviewed by:***
_Pu-Feng Du,
Tianjin University, China
Vijaykumar Muley,
National Autonomous University of Mexico, Mexico_

****Correspondence:***
_Fei Zhu
zhufei@suda.edu.cn_

***Specialty section:***
_This article was submitted to Bioinformatics and Computational Biology,
a section of the journal Frontiers in Bioengineering and Biotechnology_

***Received:*** _21 January 2020_
***Accepted:*** _07 April 2020_
***Published:*** _05 May 2020_

***Citation:***
_Li F, Zhu F, Ling X and Liu Q (2020) Protein Interaction Network Reconstruction Through Ensemble Deep Learning With Attention Mechanism.
Front. Bioeng. Biotechnol. 8:390.
doi: 10.3389/fbioe.2020.00390_

## INTRODUCTION

Protein interactions and interaction networks take part in vital activities of each living cell, including signal transduction, immune response, metabolism of energy substance, cell cycle control, etc. (Keskin et al., 2016). The exact identification of protein interactions is therefore important not only to understanding the functions of proteins but also to structure-based drug design and treatment of diseases (Li et al., 2009).

Majority of existing methods for predicting PPI are based on Gene Ontology and annotations, phylogenetic profile, gene fusion, the interacting proteins co-evolution pattern and the similarity of proteins in sequence, structure, domain and subcellular localization (Boxem et al., 2008; Zhang et al., 2012; Planas-Iglesias et al., 2013; Sun et al., 2017). However, as their accuracy and reliability depend heavily on collected prior knowledge, they are hardly applied widely. Several methods based on amino acid sequence computation have been explored to predict PPI, such as support vector machine with traditional auto-correlation, k-nearest neighbor (kNN) with local description (LD) (Yang et al., 2010), support vector machine (SVM) with conventional auto covariance (AC) (Guo et al., 2008) or local description (LD) (Zhou et al., 2011), deep neural network with amphiphilic Pseudo amino acid composition (PseAAC) descriptor (Du et al., 2017b) and so on. The above methods provide different techniques of protein sequences such as AC, LD, MCD, PseAAC, with each technique extracting different feature information of protein interactions (Zhang et al., 2019a). AC and CT considered the physical properties of amino acids and their dipole and side-chain volumes respectively. Then LD uses triples to describe composition, transition and distribution of sequence, while PseAAC further studies order information of sequences. We propose to combine different descriptors to achieve PPI prediction to obtain more information from protein interactions.

Ensemble learning is a machine learning method, which uses a series of learners and uses some rules to integrate the learning results so as to obtain better performance than a single learner. And ensemble learning has broad application prospects in many fields such as protein phosphorylation site prediction, genome function prediction and cancer prediction in bioinformatics (Gomes et al., 2017; Krawczyk et al., 2017). The previous works also demonstrate the effectiveness of classifier ensemble and provide some guidelines to generate an ensemble classification model (Martin et al., 2005; Han and Huang, 2006; Huang and Zheng, 2006; Huang and Du, 2008). Wang used a boosting technique to generate multiple classifiers iteratively to solve the problem of imbalance between positive and negative data when predicting the phosphorylation sites (Wang et al., 2017). Wang took a random forest and voting method as a basic classifier integration strategy separately to predict PPI sites (Wang et al., 2019). You et al. (2019) chose the basic classifiers with optimal performance, leaving the classifiers with small differences and using the max-wins voting (MWV) strategy to predict DNA binding proteins. Zhang et al. (2019a) trained 27 models by combining AC, MCD, LD with 9 DNN models of different configurations, and integrated these models through Double-layers BP Neural Network.

Furthermore, when exploring protein interactions and interaction networks, it is nonnegligible to quantify the interaction/non-interaction relationship between two proteins. One solution is to directly concatenate the features of the two proteins to form a feature vector (Zhang et al., 2019b), which lacks the information characteristics of the interaction/non-interaction between two proteins; another solution is to extract two features with two different networks and combine the features to form a new feature vector as the input of the model (Du et al., 2017b; Hashemifar et al., 2018), which is incapable of learning inherent relation of the proteins. Recently in natural language processing domain, researches have shown that attention mechanisms can effectively emphasize the relatively important parts of the input sentences and help boost the performance of relation extraction (Chen et al., 2017; Du et al., 2017a). In bioinformatics, attention mechanism is also adopted in chemical-protein interaction (CPI) (Zhang et al., 2019b), kinase-specific phosphorylation site prediction (Wang et al., 2017) and so on. In Xuan et al. (2019) model, exploiting the attention mechanism module to learn features or extract the relationship between IncRNA and disease provides more information. Wang et al. (2017) designed a two-dimensional independent attention mechanism for predicting phosphorylation sites, which enabled the model, called MusiteDeep, to automatically search important positions of the output sequences to estimate the contribution of each element in the sequences and feature dimensions. However, the above researches concern only single attention mechanism in the deep neural network model, which can be replaced by the multi-head attention mechanism that can exert attention multiple times and divide attention information into multiple heads. Liu et al. (2018) integrated attention pooling

**TABLE 1 |** Division of amino acid into seven groups based on the dipoles and volumes of the side chains.

| Group 1 | Group 2 | Group 3 | Group 4 | Group 5 | Group 6 | Group 7 |
|---|---|---|---|---|---|---|
| A, G, V | C | F, I, L, P | M, S, T, Y | H, N, Q, W | K, R | D, E |

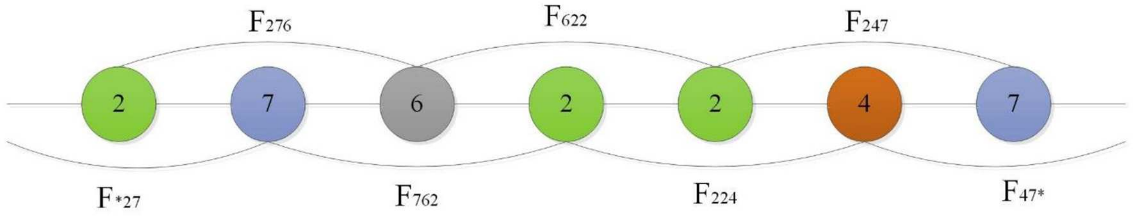

**FIGURE 1 |** For a grouped sequence “2762247,” the numerical code string of consecutive amino acids are “276,” “762,” “622,” “224,” “247,” and “$^{*}$27,” “47$^{*}$” according to Shen, and “$^{*}$” is considered to be the first or second amino acid of an amino acid in a continuous amino acid. So its triad types are F276, F762, F622, F224, F427.

into the gated recurrent unit (GRU) model to extract CPIs. Verga et al. (2018) combined the Multi-head attention with convolution neural networks to construct a transformer model to extract the document-level biomedical relations. Thus, a multi-head attention mechanism will make it easier to capture the relevant important information for deep neural networks in PPI extraction.

Motivated by attention mechanisms and ensemble learning, we propose an algorithm called EnAmDNN, which at first extracted the biophysical-chemical information of protein sequences through AC, CT, LD, and PseAAC and association with the interactive description of each protein in protein interaction network; then it automatically extracted the protein features by multi-layer convolutional neural network, adopted attention mechanism to analyze deep-seated relationship of proteins and then forms the feature vectors. In EnAmDNN, 16 kinds of DNN models are trained through 4 characteristic bases which are the inputs of 4 DNNs with different layers and different neurons. In the integration module, the outputs of 16 DNNs are taken as the inputs of deep neural networks finally, and the five-fold cross validation is adopted to comprehensively predict protein interactions and interaction networks. Our contributions can be summarized as follows: (1) the new network structure can automatically extract highly abstract representations and detect the sequence specificity of proteins; (2) the attention mechanism is adopted to analyze internal links between the two proteins and the network description of each protein, instead of directly concatenating the two proteins, to improve the prediction accuracy; (3) ensemble learning considers the advantages of different descriptors and different DNNs to achieve comprehensive learning.

# PRELIMINARIES

## Deep Neural Network

It turns out that deep neural network (DNN) plays an important role in bioinformatics (Alipanahi et al., 2015; Zhou and Troyanskaya, 2015; Liu et al., 2016), i.e., predicting inner-organization and trans-organization RNA splicing patterns (Leung et al., 2014). DeepMind applied DNN to the detection of sequence specificity of the DNA-RNA binding protein, which is superior to other methods (Alipanahi et al., 2015); DeepSEA applied DNN to learn the code of regulatory sequences from chromatin map sequences in order to discern priorities of other functional varieties (Zhou and Troyanskaya, 2015); other examples include genome informatics extraction, detection of protein structure and medicine discovery. In short, compared with other sequence-based methods, DNN has the following advantages: (1) it can automatically learn certain protein sequences; (2) it can reduce the influence of noise on the raw data and extract the hidden high dimension representation (Bengio et al., 2013). However, the performance of DNN is closely related to the network configuration and may vary greatly for different configurations.

## Protein Representation Technique

Different representation techniques of protein features may have a strong impact on the performance of PPI prediction, making it a challenge to effectively express the protein features and describe the connections of two proteins. We choose four representative protein techniques instead of one to avoid the limitation brought by a single technique.

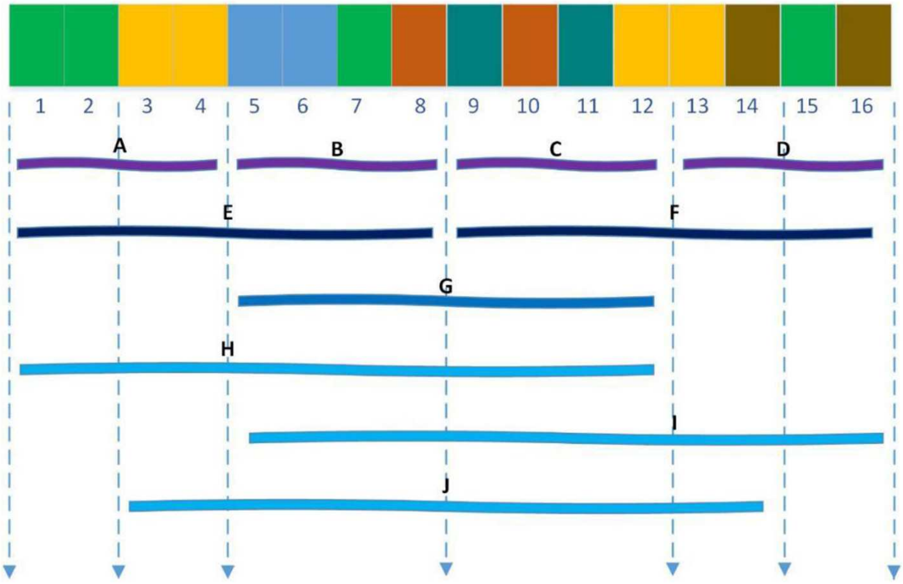

**FIGURE 2 |** Ten regions **(A–J)** of the entire protein sequence. The regions **(A–D)** are generated by dividing the whole sequence into four equal regions, and regions **(E,F)** are generated by dividing the whole sequence into two equal regions. The region **(G)** stand for the central 50% of the entire sequence. And the regions **(H–J)** stand for the first 75%, the final 75% and the central 75% of the entire sequence respectively.

## Auto Covariance Technique

Two proteins interact with each other through electrostatic, hydrophobic, steric and hydrogen bond, which can be reflected by the seven physicochemical properties of amino acids, including hydrophobicity ($H_1$), hydrophilicity ($H_2$), volumes of side chains of amino acids (VSC), polarity ($P_1$), polarizability ($P_2$), solvent-accessible surface area and net charge index of side chains. The above properties are exploited by the auto-covariance method to transform amino acid sequence into uniform matrices which reveal the special connection of two residues under a certain distance and are widely applied in protein-encoding. For example, a protein sequence of length $L$ is calculated as follows (Guo et al., 2008):

$$AC(lag, j) = \frac{1}{L - lag} \sum_{i=1}^{L-lag} (X_{i,j} - \frac{1}{L} \sum_{i=1}^{L} X_{i,j}) \times (X_{(i+lag),j} - \frac{1}{L} \sum_{i=1}^{L} X_{i,j}) \quad (1)$$

*Xij* represents the *j*-th physical property of the *i*-th amino acid in the protein sequence; *lag* represents the distance between residues; then proteins of various lengths are encoded as vectors of equal length *lg* * p, where lg is the maximum *lag* (lag = 1, 2, ..., lg), p is the number of physical properties. In this study, p was 7, reflecting the characteristics of the seven amino acids. As with Guo, we set the log to 30 (Guo et al., 2008). Therefore, each protein sequence is represented as a 210-dimensional vector.

## Conjoint Triad Technique

Shen et al. (2007) introduced a conjoint triad technique to represent sequence information of each protein, in which any three contiguous amino acids are regarded as a unit and the characteristics of one amino acid and its vicinal amino acids are fully considered. First, the conjoint triad divides 20 standard amino acids into 7 groups according to their dipole and side-chain volumes, then the triads can be distinguished according to the type of amino acid. According to Shen's settings, there are 343 (7 × 7 × 7) triad types (Shen et al., 2007), as shown in **Figure 1**.

Finally, the PPI information of protein sequences are projected into the homogeneous vector space according to the

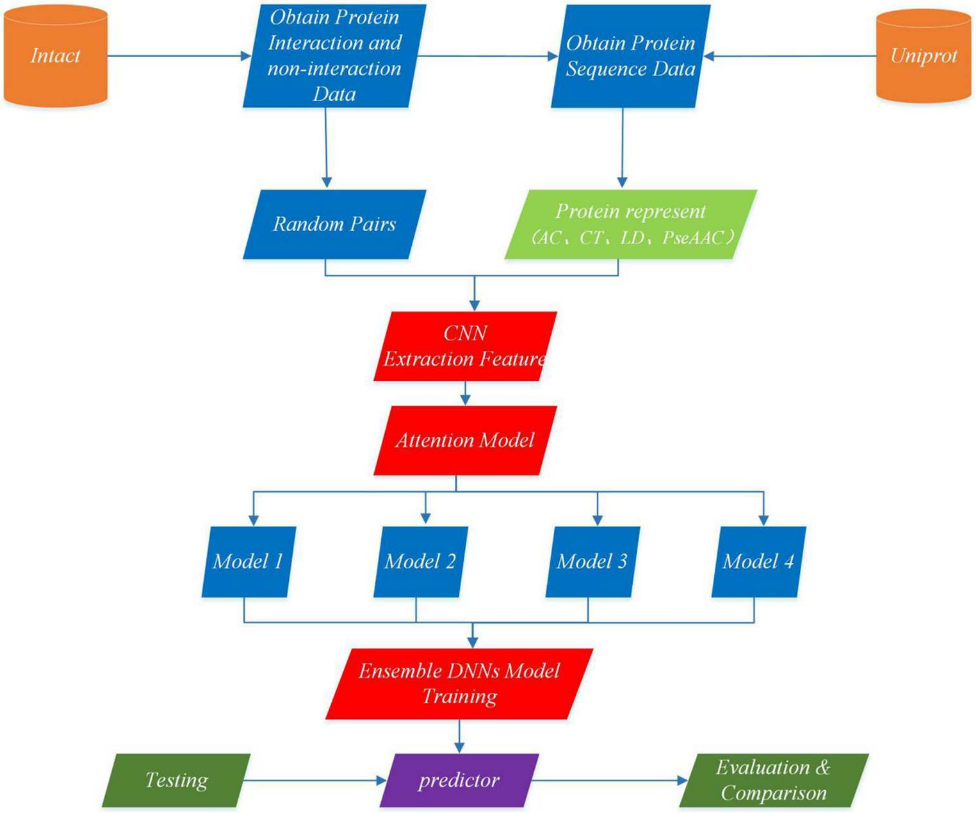
Intact
Obtain Protein Interaction and non-interaction Data
Obtain Protein Sequence Data
Uniprot
Random Pairs
Protein represent (AC、CT、LD、PseAAC)
CNN Extraction Feature
Attention Model
Model 1
Model 2
Model 3
Model 4
Ensemble DNNs Model Training
Testing
predictor
Evaluation & Comparison

**FIGURE 3 |** Flowchart of EnAmDNN for predicting protein-protein interactions. First, the interaction pairs and non-interaction pairs of related proteins are obtained from IntAct, and all protein sequence data of UniProt are obtained; the appropriate proportion of interaction pairs and non-interaction pairs are selected, and each group of protein pairs (including interaction pairs and non-interaction pairs) is vectorization by AC, CT, LD, and PseAAC techniques; put vector protein into convolution neural network for feature extraction of each protein; extracted features are transferred to attention mechanism module for deep analysis of interaction between each group of protein pairs; then the analyzed features are input into deep neural network of different models for training; finally, the final prediction results are obtained by integrating the prediction results of different models.

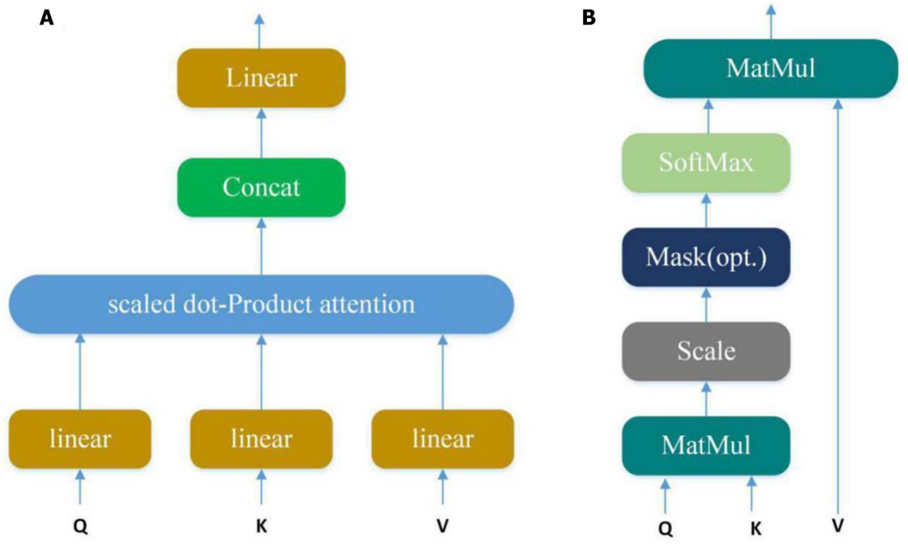

**FIGURE 4 | (A)** Multi-Head Attention consists of several attention layers. (Vaswani et al., 2017) First, query, key and value go through a linear transformation, and then enter them into scaled dot-Product attention to generate many heads; then concatenate these heads to keep relevant information in different representation subspaces. **(B)** Scaled dot-product attention (Vaswani et al., 2017). Obtain weights by similarity calculation between query and each key, and the weights are normalized by softmax function; then attention is obtained by the weight and the corresponding value.

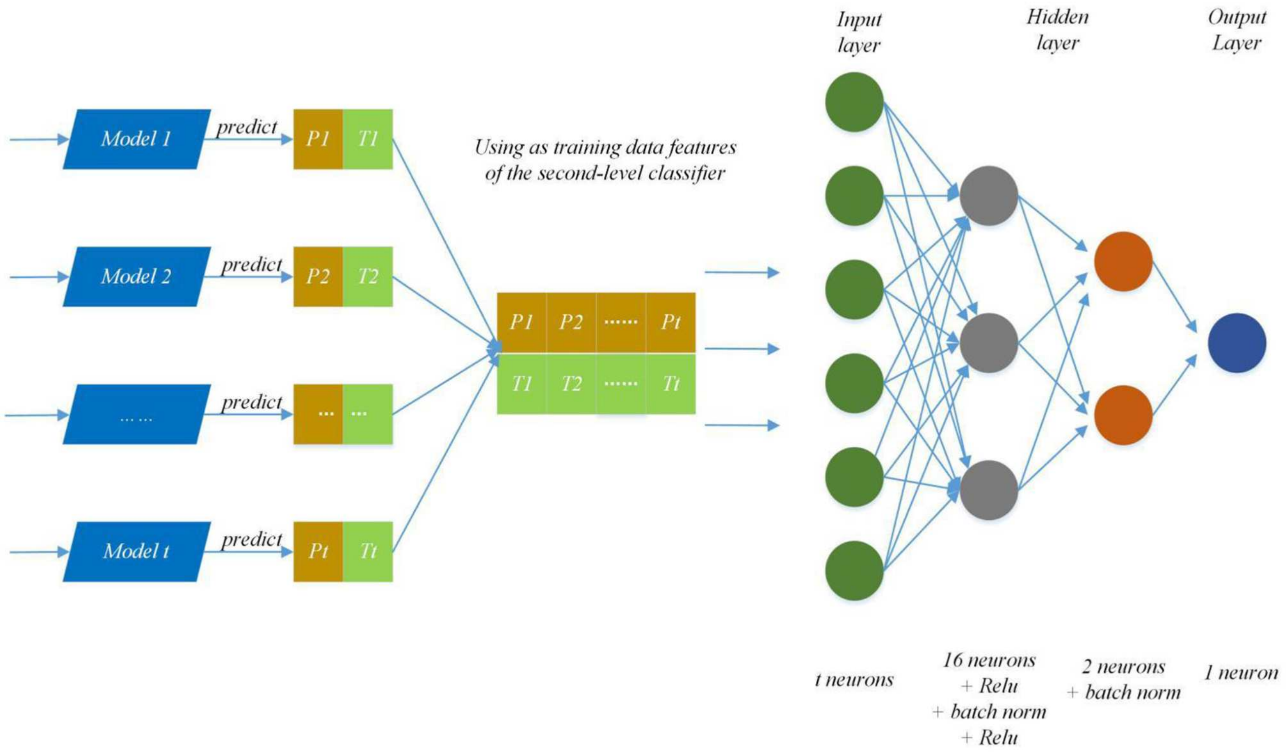

**FIGURE 5 |** Ensemble strategy composed of deep neural networks. The first layer results ($P_n$, $T_n$) ($0 < n < t+1$) are predicted by T primary learners, where $P_n$ and $T_n$ stand for training data and the prediction result; then use recombined ($P_n$, $T_n$) as training data features of the second-level classifier and put it into deep neural networks to predict protein interaction networks.

frequency of each triad type, where each protein is represented by a 343-dimensional vector.

### Local Descriptor Technique

The Local descriptor technique (Zhou et al., 2011) also divided 20 standard amino acids into 7 groups as shown in **Table 1** and divided the entire protein sequence into 10 regions as shown in **Figure 2**. For each sub-sequence, three descriptors, composition (C), transition (T) and distribution (D), are applied to describe its trait where C represents the proportion of each amino acid group; T represents the frequency with which amino acids in one group are followed by amino acids of another group; D measures the proportion of chain length where the top 25, 50, 75, and 100% of the amino acids of a particular group are located. For the local descriptor method, each region produces 63 values, where C represents 7, T represents 7, and D represents 35, and then each protein is encoded as a 630-dimensional vector.

For example, according to **Table 1**, the sequence "ACLACLCCLAALLCCCLALALAAALL" is converted into the amino acid group "12312322311332321311313111133133" so that the sub-sequence contains 9 "1", 7 "2," and 10 "3."For feature C, 9/(9 + 7 + 10) = 0.3461, 7/(9 + 7 + 10) = 0.2693, 10/(9 + 7 + 10) = 0.3846; for feature T, there are 2 cases that "1" is converted to "2" or "2" is converted to "1," then 2/25 = 0.08; similarly, transitions between "3" and "1" as well as "2" and "3" are 3/25 = 0.12, 6/25 = 0.24, respectively; for feature D, there are nine "1"s, then the D descriptor for 1 is 1/26 = 0.0384, [0.25*9 + 0.5]/26 = 0.0769, [0.5*9 + 0.5]/26=0.1923, [0.75*9 + 0.5]/26 = 0.2692, [1*9 + 0.5]/26 = 0.3462.

### Pseudo Amino Acid Composition (PseAAC) Technique

Tian et al. (2019) used a sequence encoding technique based on pseudo amino, that is, PseAAC. Given a protein sequence P with *L* amino acid residues:

$$S_1S_2S_3S_4.......S_L$$

where $S_i$ represents the *i*th residue of the protein P, $1\leq i\leq L$.

According to the PseAAC technique, the protein P can be formulated as

$$P = [x_1, x_2, ..., x_{20}, x_{21}, ..., x_{20+\lambda}]^T,\ (\lambda < L) \tag{2}$$

where the 20 + λ components are given by

$$x_k = \begin{cases} \dfrac{f_k}{\sum\limits_{i=1}^{20} f_i + \omega \sum\limits_{j=1}^{\lambda} \theta_j}, (1 \leq k \leq 20) \\ \dfrac{\omega\theta_{k-20}}{\sum\limits_{i=1}^{20} f_i + \omega \sum\limits_{j=1}^{\lambda} \theta_j}, (21 \leq k \leq 20 + \lambda) \end{cases} \tag{3}$$

**TABLE 2 |** AD with various methods.

| Methods | Accuracy | Recall | Precision | F1 | AUC |
|---|---|---|---|---|---|
| EnsAmDNN | **0.9467** | **0.9329** | **0.9541** | **0.9433** | **0.9463** |
| SVM_AC | 0.7886 | 0.6369 | 0.8956 | 0.7435 | 0.7831 |
| kNN_LD | 0.7549 | 0.7507 | 0.7602 | 0.7542 | 0.7556 |
| SVM_LD | 0.804 | 0.6735 | 0.9174 | 0.7754 | 0.805 |
| NDDs_APAAC | 0.898 | 0.9093 | 0.8907 | 0.8993 | 0.8977 |
| EnsDNN | 0.9372 | 0.9255 | 0.9531 | 0.9388 | 0.938 |

*The highest score of each evaluation criteria is emphasized in bold.*

**TABLE 3 |** PD with various methods.

| Methods | Accuracy | Recall | Precision | F1 | AUC |
|---|---|---|---|---|---|
| EnsAmDNN | **0.895** | **0.8568** | 0.9275 | **0.8903** | **0.8951** |
| SVM_AC | 0.797 | 0.62 | 0.9373 | 0.7444 | 0.7904 |
| kNN_LD | 0.8057 | 0.8042 | 0.7906 | 0.7958 | 0.8064 |
| SVM_LD | 0.8315 | 0.7337 | 0.9084 | 0.8108 | 0.831 |
| NDDs_APAAC | 0.8773 | 0.8247 | 0.9058 | 0.8627 | 0.8906 |
| EnsDNN | 0.8917 | 0.8433 | **0.9311** | 0.8846 | 0.8915 |

*The highest score of each evaluation criteria is emphasized in bold.*

**TABLE 4 |** Cancer with various methods.

| Methods | Accuracy | Recall | Precision | F1 | AUC |
|---|---|---|---|---|---|
| EnsAmDNN | **0.8502** | **0.8062** | **0.8863** | **0.8436** | **0.8508** |
| SVM_AC | 0.6524 | 0.4848 | 0.7347 | 0.5811 | 0.6545 |
| kNN_LD | 0.6475 | 0.6761 | 0.6202 | 0.6458 | 0.6471 |
| SVM_LD | 0.6673 | 0.5591 | 0.715 | 0.6263 | 0.6681 |
| NDDs_APAAC | 0.7551 | 0.7224 | 0.7764 | 0.7474 | 0.7555 |
| EnsDNN | 0.8008 | 0.7549 | 0.8362 | 0.7925 | 0.802 |

*The highest score of each evaluation criteria is emphasized in bold.*

**TABLE 5 |** Cancer with various methods.

| Methods | Accuracy | Recall | Precision | F1 | AUC |
|---|---|---|---|---|---|
| EnsAmDNN | **0.907** | 0.8523 | **0.96** | **0.9018** | **0.9088** |
| SVM_AC | 0.7356 | 0.6316 | 0.806 | 0.7037 | 0.7371 |
| kNN_LD | 0.7671 | 0.7246 | 0.7934 | 0.7565 | 0.7675 |
| SVM_LD | 0.7819 | 0.7545 | 0.7962 | 0.774 | 0.7816 |
| NDDs_APAAC | 0.8454 | 0.8326 | 0.837 | 0.8339 | 0.8446 |
| EnsDNN | 0.9039 | **0.8747** | 0.9252 | 0.899 | 0.9034 |

*The highest score of each evaluation criteria is emphasized in bold.*

**TABLE 6 |** Diabetes with various methods.

| Methods | Accuracy | Recall | Precision | F1 | AUC |
|---|---|---|---|---|---|
| EnsAmDNN | **0.8333** | 0.871 | 0.7941 | **0.8308** | **0.8355** |
| SVM_AC | 0.7891 | 0.6667 | 0.7941 | 0.7128 | 0.7819 |
| kNN_LD | 0.7568 | 0.8571 | 0.75 | 0.8 | 0.7411 |
| SVM_LD | 0.7813 | 0.8269 | 0.7361 | 0.7742 | 0.7885 |
| NDDs_APAAC | 0.803 | **0.9118** | 0.7561 | 0.8267 | 0.7996 |
| EnsDNN | 0.8030 | 0.7576 | **0.8333** | 0.7937 | 0.8030 |

*The highest score of each evaluation criteria is emphasized in bold.*

In Equation (3), $f_k(k = 1, 2, \ldots, 20)$ are the normalized occurrence frequencies of 20 amino acids in protein P; $\omega$ is the weighting factor set to 0.05 in general work; and $\theta_j(j = 1, 2, \ldots, \lambda)$ denotes the order relationship between two residues that are $j$ residues apart, which is shown as follows:

$$\theta_j = \frac{1}{L-j}\sum_{i=1}^{L-j} J_{i,i+j}(j < L) \tag{4}$$

$$J_{i,i+j} = \frac{1}{3}\sum_{p=1}^{3}\left[H_p(A_{i+j}) - H_p(A_i)\right] \tag{5}$$

where $J_{ij}$ denotes the order relationship function between amino acid $A_i$ and $A_j$, $H_p(A_i)$ denotes the $p$th property of $A_i$. $H_1(A_i)$, $H_2(A_i)$ and $H_3(A_i)$ are the hydrophobicity value, hydrophilicity value and side-chain mass for the amino acid, respectively. This coding method contains more sequence characteristics because it considers not only the physicochemical properties of the protein but also the order information of sequences.

# MATERIALS AND METHODS

## Data Sets

We collect the dataset information of Parkinson's disease (PD), Alzheimer's disease (AD), cancer, cardiac and diabetes, whose interactive information is from IntAct database (Kerrien et al., 2007) and sequence information from Uniprot (Bairoch et al., 2004). We are concerned about positive-negative selection in our work. For the positive set, proteins and protein pairs that contain less than 50 amino acids and 40% of sequence identity are removed to eliminate the variance caused by minor bias proteins to the model. The negative set was obtained by pairing proteins whose subcellular localization is different (Guo et al., 2008) or GO Cellular Component (CC) and Biological Process (BP) ontology with experimental evidence codes (Muley and Ranjan, 2012). The subcellular location information on the proteins is extracted from Uniprot. According to this information, a protein can be divided into several types of localization cytoplasm, nucleus, mitochondrion, endoplasmic reticulum, Golgi apparatus, peroxisome and vacuole. The way to construct negative set must meet the following requirements: (1) the non-interacting pairs cannot appear in the positive data set; (2) the contribution of proteins in the negative set should be as

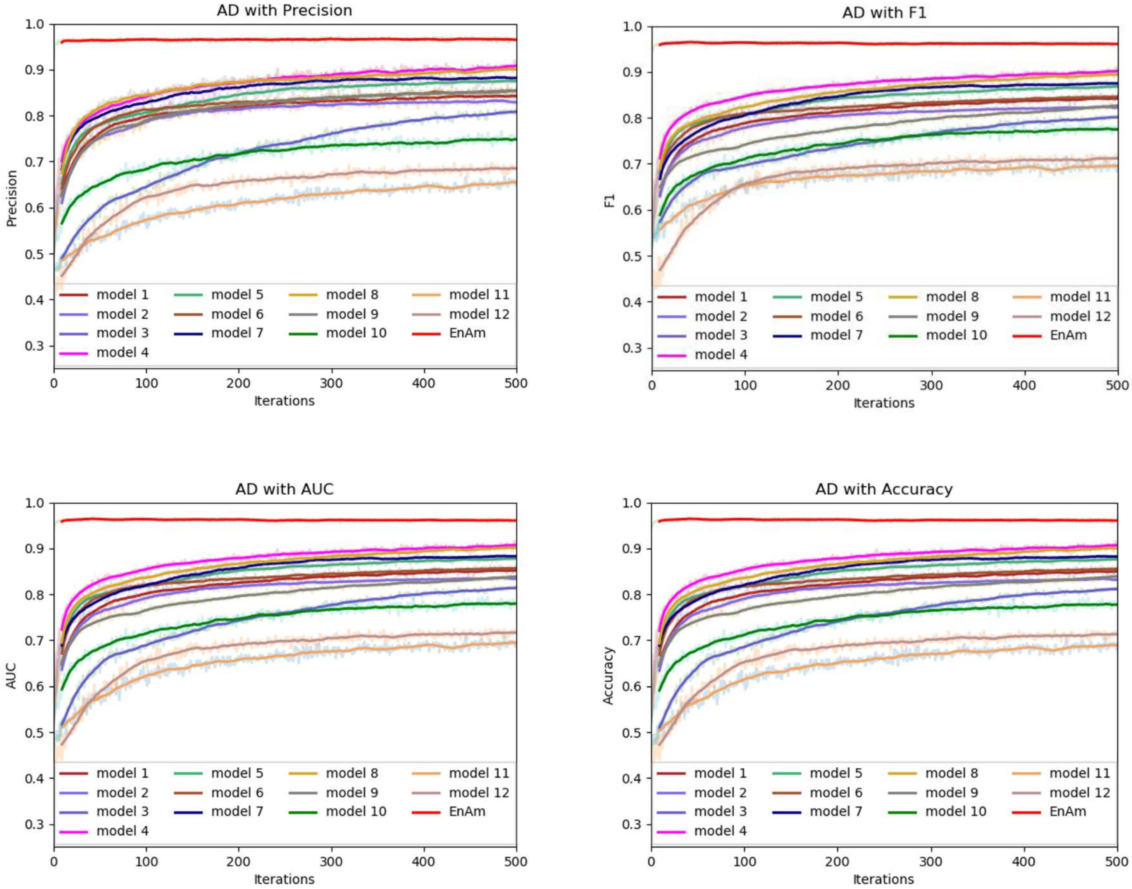

**FIGURE 6 |** Comparison of evaluation indexes of each basic model and ensemble model with AD data set.

harmonious as possible. In our work, the ratio between positive and negative set is 1:1, where the negative sets are randomly chosen from non-interactive pairs.

Finally, we have five independent PPI datasets: Parkinson's disease (PD) (4,127 interacting pairs and 4,127 non-interacting pairs), Alzheimer disease (AD) (4,096 interacting pairs and 4,096 non-interacting pairs), Cancer (6,352 interacting pairs and 6,352 non-interacting pairs), Cardiac (2,663 interacting pairs and 2,663 non-interacting pairs) and Diabetes (163 interacting pairs and 163 non-interacting pairs).

## Evaluation Criteria

The following metrics are taken into account to perform evaluation: Overall Prediction Accuracy, Recall, Precision, $F_1$ score values, and Area under the ROC Curve (AUC) (Zhang et al., 2019a). The first four metrics are defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{8}$$

$$F_1 = \frac{2TP}{2TP + FP + FN} \tag{9}$$

where TP (true positive) is the number of true interacting pairs found in the positive data set, TN (true negative) is the number of true non-interacting pairs with correct prediction, FP (false positive) is the number of the predicted interacting pairs not found in the positive data set, and FN (false negative) is the number of the true interacting pairs with false prediction.

## Ensemble Deep Neural Networks

This section describes EnAmDNN model that predicts PPI based on protein sequences, which consists of the input module, the convolution module, the attention mechanism module, the DNN module and the integration module. Each protein sequence is encoded, by the input module, through the protein representation techniques, as a vector, whose feature is extracted by the convolution module. Then, the internal link in the protein pair is detected through the attention module, and then the analyzed protein pair is provided to 16 dependent learners. After the training is completed, these learners will be integrated

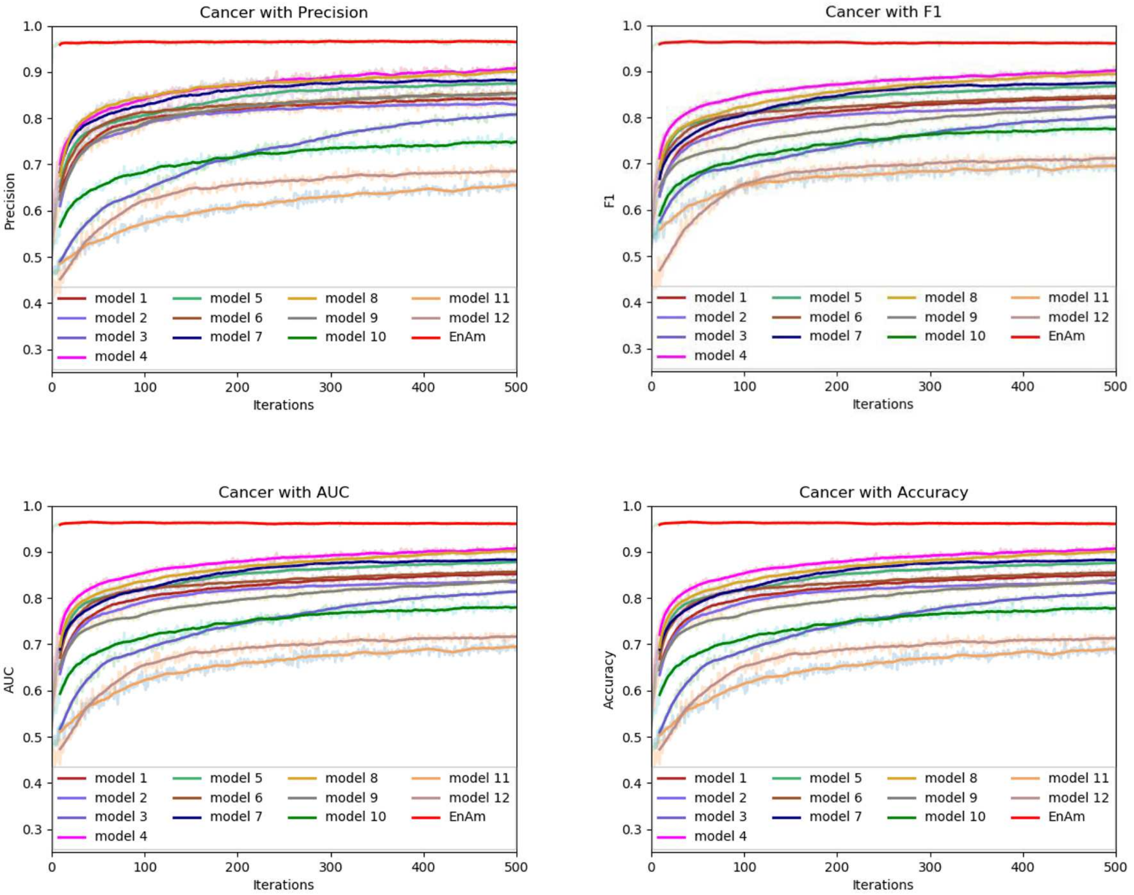

**FIGURE 7 |** Comparison of evaluation indexes of each basic model and ensemble model with Cancer data set.

through a two-layer hidden layer neural network. The working process of an EnAmDNN is shown as **Figure 3**.

## Deep Convolutional Module

The convolution module is a batch of normalized layers, a stack of convolutional layers and activation layers, which can automatically extract features of vectorized protein sequences. In our model, the output of the convolution module is calculated by an expression that starts with a convolution layer and ends with a convolution layer:

$$In_{t+1} = RELU\left(Batch_{\beta,\delta}\left(Conv_{\lambda}\left(In_t\right)\right)\right) \tag{10}$$

$$Out = Conv_{\gamma}\left(In_n\right) \tag{11}$$

Repeat $In$ n times and enter the result $In_n$ into Equation 11. Where $Out$ is the output vector, $In$ is the input vector and β, γ, δ, λ are the parameters of batch normalization and convolution layers.

The convolution layer searches the sequences according to their input orders and outputs the corresponding features; the batch normalization layer takes in the feature vectors and normalizes their mean values and the variances; ReLU layer takes in the normalized vectors and introduces non-linearity to achieve complex function approximation. Then the above processes repeat n times to obtain the feature vector.

## Attention Module

Convolution layer can automatically learn potential features from input sequences, but only a small part of these potential features are very important in PPI. In our model, we use the multi-attention mechanism to adjust the weight of the input sequences to further emphasize the relatively crucial features. Applying the attention multiple times may learn more important features than single attention and allowing the model to learn relevant information in different representation subspaces (Vaswani et al., 2017). It can be understood that attention selectively selects a small amount of important information that is beneficial to PPI from a large amount of information and focuses on important information, ignoring most of the insignificant information. We choose the mechanism of multiple attention rather than directly connecting the two protein eigenvectors to increase the exploration of protein pairs and further use the indirect relationship between residues to obtain more accurate information. The Multi-head attention calculates the output based on the query and a set of key-value pairs, where $Q$, $K$, $V$ denote query, key, and value respectively. The specific structure is shown in **Figure 4**:

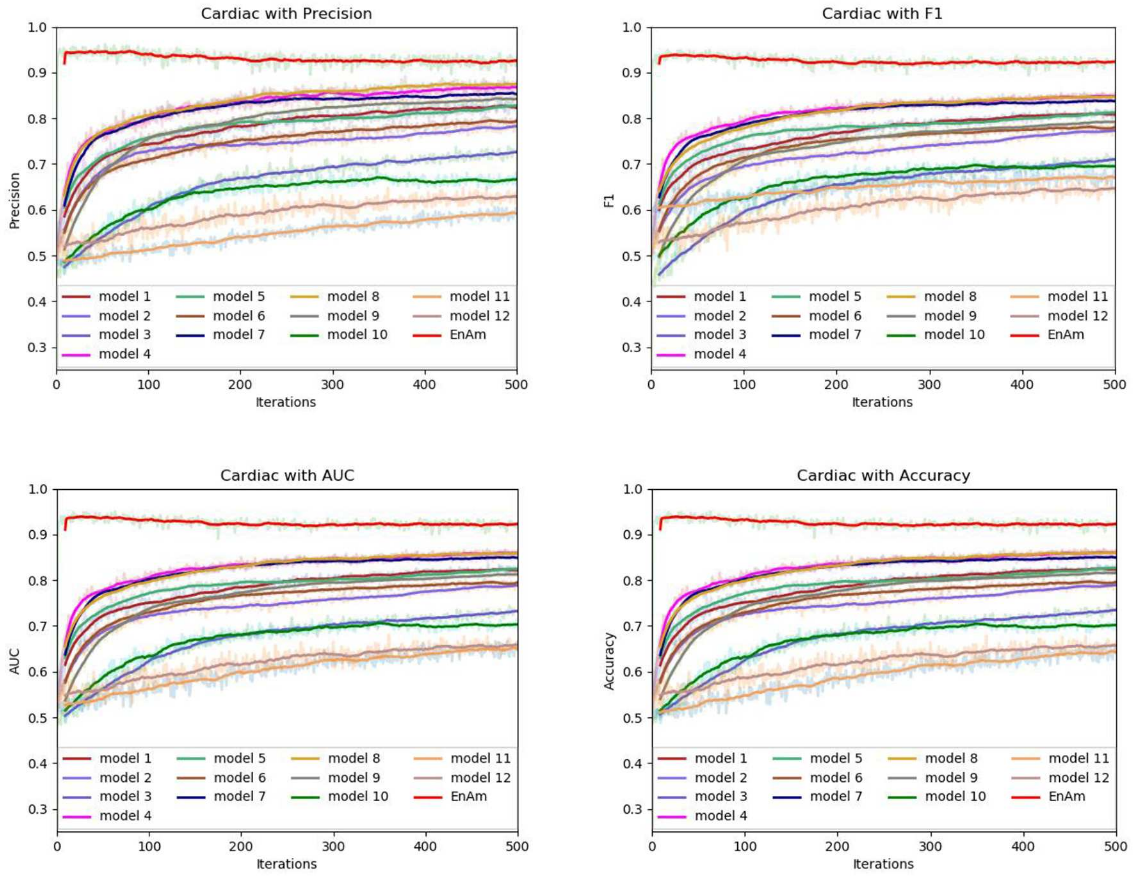

**FIGURE 8 |** Comparison of evaluation indexes of each basic model and ensemble model with Cardiac data set.

Query, key and value go through a linear transformation first, and then enter them into scaled dot-Product attention. At this time, the attention calculation formula is as follows:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK'}{\sqrt{d_k}})V \tag{12}$$

Where $\sqrt{d}$ is scaling factor. The core of the Multi-head attention is employing the above attention multiple times, and one time attention means one head. Suppose the Multi-head attention needs to be done $h$ times to generate $h$ heads, the $\mathrm{Att_{head}}$ can be calculated as follows:

$$Att_{headi}(Q, K, V) = Attention(QW_i^Q, KW_i^K VW_i^V), 1 < i \leq h \tag{13}$$

where $W_i^Q, W_i^K, W_i^V$ are parameter matrices. Finally, these heads are concatenated and once again linearly transformed by

$$MultiHead(Q, K, V) = Linear(Concat(Att_{head1}, ..., Att_{head\mathrm{h}})W^{\mu}) \tag{14}$$

In order to keep the invariance of features, we introduce average pooling and maximum pooling to reduce the errors caused by model parameters and retain information of global and local features.

$$\begin{aligned} &newMultiHead(Q, K, V) \\ &= Concat(AvgPool(MatMul(MultiHead(Q, K, V), Q)), \\ &MaxPool(MatMul(MultiHead(Q, K, V), Q))) \end{aligned} \tag{15}$$

where *AvgPool* is the function of average pooling and *MaxPool* is the function of maximum pooling.

For a protein pair ($\mathrm{P_1, P_2}$), it is expressed as $S_1, S_2$ respectively after convolution layer. We use the merge layer to fuse the protein pairs that are redistributed by the attention mechanism. The calculation formula of the merge layer is as follows:

$$\begin{aligned} S_1{}' &= newMultiHead(S_1, S_2, S_1), \\ S_2{}' &= newMultiHead(S_2, S_1, S_2) \end{aligned} \tag{16}$$

$$Merge(S_1{}', S_2{}') = Concat(\frac{S_1{}' \cdot S_2{}'}{|S_1{}'| \times |S_2{}'|}, S_1{}'AS_2{}', S_1{}', S_2{}') \tag{17}$$

where $A$ is weight.

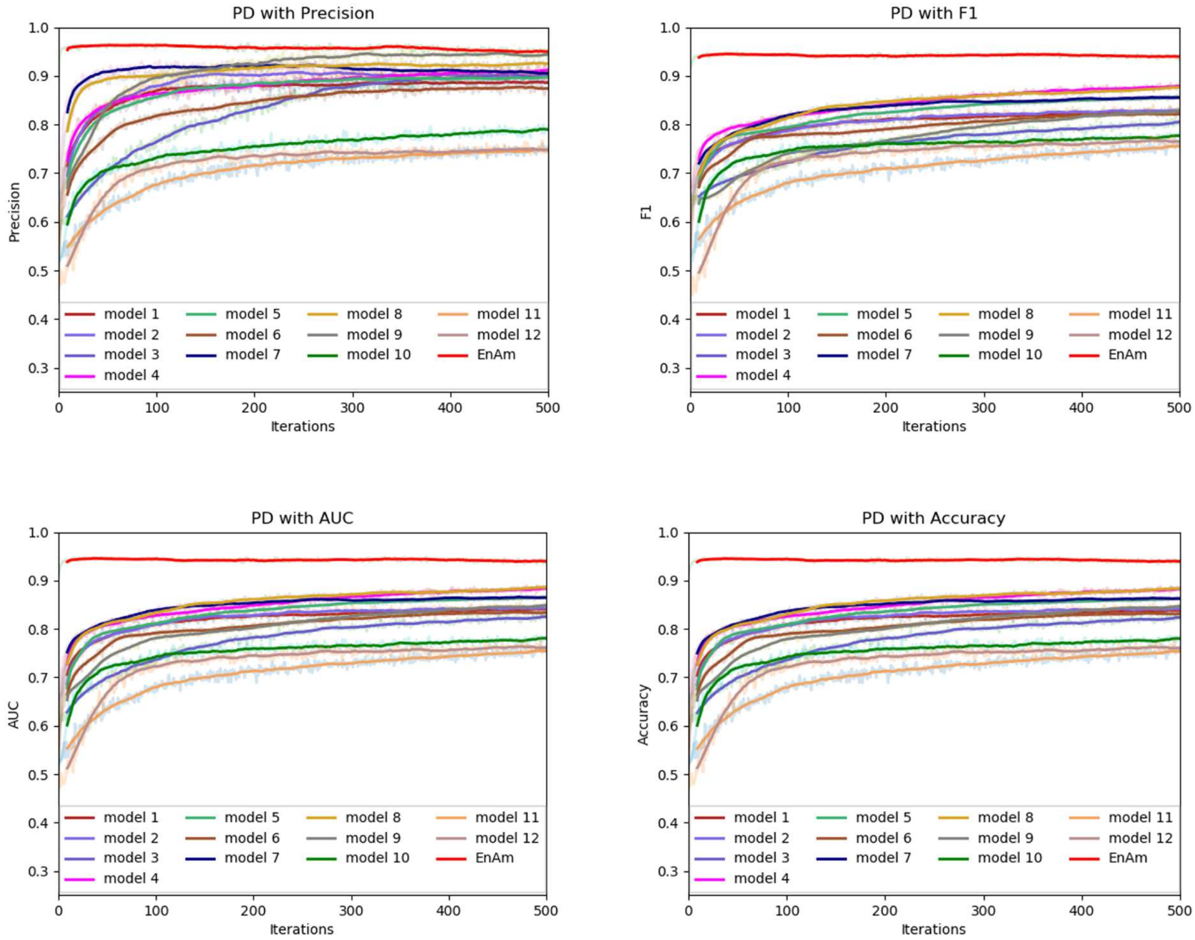

**FIGURE 9 |** Comparison of evaluation indexes of each basic model and ensemble model with PD data set.

The basic model of our work is mainly constructed by 3.3.1 the deep convolution module, 3.3.2 the attention mechanism module and the deep neural network. The specific basic learning algorithm is shown in algorithm 1.

**Algorithm 1:** Basic learning algorithm based on multi-head attention with pooling

**Input:**interaction data $D = \{x_i, y_i\}_{i=1}^{n}$, encoded protein sequence pair$x_i = (s_{i1}, s_{i2})$
**Return:** basic learning algorithm parameters W, b

```
for $i = 1$ to $n$:
    $S_1, S_2 \leftarrow x_i(s_{i1}, s_{i2})$ represent encoded protein $p_1$ and protein $p_2$
    for $iter = 1$ to $t$:
        $S_{1,iter+1} = RELU\left(Batch_{\beta,\delta}\left(Conv_{\lambda}\left(S_{1,iter}\right)\right)\right)$
        $S_{2,iter+1} = RELU\left(Batch_{\beta,\delta}\left(Conv_{\lambda}\left(S_{2,iter}\right)\right)\right)$
    end for
    $S_1 \leftarrow Conv_{\gamma}\left(S_{1,t}\right)$
    $S_2 \leftarrow Conv_{\gamma}\left(S_{2,t}\right)$
    calculate the importance value of single attention by
        $Attenhead_{S_1}^{i} = \mathrm{softmax}\left(\frac{S_1 S_2'}{\sqrt{d_k}}\right) S_1,$
        $Attenhead_{S_2}^{i} = \mathrm{softmax}\left(\frac{S_2 S_1'}{\sqrt{d_k}}\right) S_2$
    connect multiple attentions by $S_1^{*} = MultiHead(S_1, S_2, S_1)$,$S_2^{*} = MultiHead(S_2, S_1, S_2)$
    Calculate new feature importance value of multi-head attention with pooling by
        $S_1^{**} = newMultiHead(S_1^{*}, S_2^{*}, S_1^{*}),$
        $S_2^{**} = newMultiHead(S_2^{*}, S_1^{*}, S_2^{*})$
    merge the sequence feature pairs as the input of the network by $Concat(\frac{S_1^{**} \cdot S_2^{**}}{|S_1^{**}| \times |S_2^{**}|}, S_1^{**} A S_2^{**}, S_1^{**}, S_2^{**})$
    $output = \mathrm{softmax}\left(W \cdot merge + b\right)$
end for
Return $W, b$
```

### Ensemble Strategy

Ensembles of independent deep neural networks can improve the performance of a single network (Bairoch et al., 2004). In the Otto group product classification challenge, the first one won the championship by stacking 30 models. The model achieved remarkable results, and we also adopted the stacking method of ensemble learning in our work. Secondly, to predict the effect better, the trainers of each primary model keep stability and diversity as much as possible.

We modify the internal structure of the algorithm and learn from different feature representations, which are two strategies to maintain diversity and achieved improvement, so we also take the same measures (Zhang et al., 2019a). In practice, we choose four feature representations to quantify the characteristics of each protein and set different parameters of DNNS according to the characteristics of each representation. Then we use the stacking method to combines with five-fold cross validation, the primary learners are trained from the initial data set, and a new data set is generated by the primary learners for training the secondary learner. It means the output of each primary trainer is input as an example to the secondary trainer for fusion output and PPI prediction. Here, the secondary trainer is composed of deep neural networks. Its structure is shown in **Figure 5** and ensemble strategy is described in algorithm 2.

**TABLE 7** | Comparison of EnAm-Con and EnAm-Sep.

| Data sets | EnsAC | EnsLD | EnsCT | EnsPseAAC | EnsCon | EnAMDNN |
|---|---|---|---|---|---|---|
| AD | 0.8502 | 0.9029 | 0.8949 | 0.7759 | 0.9338 | **0.9456** |
| PD | 0.8378 | 0.8788 | 0.8820 | 0.8059 | **0.9242** | 0.8951 |
| Cancer | 0.7638 | 0.7911 | 0.7889 | 0.6928 | 0.8468 | **0.8508** |
| Cardiac | 0.8201 | 0.8789 | 0.8888 | 0.7455 | 0.9020 | **0.9088** |
| Diabetes | 0.7516 | 0.7576 | 0.7917 | 0.6921 | 0.8459 | **0.8787** |

*The highest score of each evaluation criteria is emphasized in bold.*

# RESULT

## Comparing the Prediction Performance With Other Methods

All the experiment were carried out on a computer with CentOs, Cuda version 10.1.243, CuDnn version 7.0 and software environment python3.7+keras2.0+torch1.3.

In order to evaluate the performance of EnAmDNN, we compared it with the approaches proposed by Guo et al. (2008), Zhou et al. (2011), Du et al. (2017b), Yang et al. (2010), Zhang et al. (2019a) emphasized the highest score of each evaluation criteria in bold and present the results in **Tables 2–6**, which separately utilize AC, LD, CT, APAAC, PseAAC to encode amino acid sequence, and predicted PPI with SVM, k-nearest neighbor (kNN) or DNNs, all of which share the same training sets and the same testing sets. It can be seen from **Table 2** that EnAmDNN generally outperforms these predictors, where EnAmDNN achieved optimal prediction performance in all the datasets, especially in AD, with an accuracy of 94.67%, and a recall rate of 93.29%. The accuracy is 95.41%, F1 is 94.33%, and AUC is 94.63%. This is because, in EnAmDNN, feature representations in protein sequences are coordinated, and new features are obtained through different classifiers. Compared with the recent EnsDNN model, in five independent data sets, the AUC index DnAmDNN has increased by 0.89, 0.41, 0.61, 0.6, 3.90%, and the accuracy of PPI prediction are relatively high. The EnAmDNN model takes advantage of the multi- head attention mechanism, that is, extracts the internal links of the PPI, thereby improving the performance of the model.

To further demonstrate the effect of ensemble strategy, five-fold cross-validation is employed to improve the reliability of the results. **Figure 6** shows the performance of each basic learner, where it can be observed, taking AD dataset as an example,

**Algorithm 2:** EAM algorithm

**Input:** interaction data; basic learning algorithm $\psi_1, \psi_2, \ldots, \psi_M$; L layers; learning rate $\eta$; Max iterations and iteration threshold $\varepsilon$; class k$\in$ {0, 1}

**Return:** ensemble strategy parameters *W, b*

```
initialize weight matrix $W$ and Bias $b$, $iter$ as 1
extract protein sequence from uniprot
vectorize protein sequence by AC, LD, CT, PseAAC
divide training data $TrD = \{D_i\}_{i=1}^{K} = \{x_i, y_i\}_{i=1}^{n}$ and testing data $TeD$
for $m = 1$ to $M$ do
    for $k = 1$ to $K$ do
        train $h_m = \phi_m(\bar{D}_k)$ with $\bar{D}_k$
        predict $D_k$ and require predicted value $P_{mk} = h_m(D_k)$
        predict $T$ and and require predicted value $T_k = h_m(T)$
    end for
    splicing predicted value of training data through $m$th basic model by $P_m = (P_{m1}, P_{m2}, ..., P_{mK})^T$
    splicing predicted value of testing data through $m$th basic model by $T_m = (T_1 + T_2 + ... + T_K)/K$
end for
construct new training data by
and construct new testing data by $newT = (T_1, T_2, ..., T_M)$
Repeat:
    $iter$+1
    for i = 1 to $n$:
        set $a^{(1)} = x_i^*$
        for $l$=2 to $L$: $a^l = \sigma(W^l a^{l-1} + b^l)$
        calculate probability $p_k$ divided into class k
        calculate $loss^L = -\sum_{k=0}^{1} y_k^L \cdot \log(p_k)$
        for $l$=2 to $L$:
        end for
        for $l$=2 to $L$:
            $\nabla W^l = \mu \sum_{i=1}^{n} loss^l (a^{l-1})^T, \nabla b^l = \mu \sum_{i=1}^{n} loss^l$
            $W^l \leftarrow W^l - \nabla W^l, b^l \leftarrow b^l - \nabla b^l$
        end for
Until $\nabla W, \nabla b < \varepsilon$
Return $W$, $b$
```

that each basic learner, associated with five-fold cross-validation method, shows fairish prediction performance, which is reflected on all the evaluation criteria Accuracy, Recall, Precision, F1, AUC. The result indicates that our model extracts and trains the features produced by basic learners and that the shortcoming of each basic learner is overcome to a certain degree. It's also confirmed with PD, Cardiac and Cancer in **Figures 7–9**.

## Performance of PPI Prediction

To further study the effectiveness of the ensemble strategy, we designed two different network architectures: (a) concatenating four feature representations (AC, LD, CT, PseAAC) as the input to the first layer classifiers (namely EnAm-Con) and (b) separately taking one feature representation as the input to the first layer classifiers (namely EnAm-Sep including EnsAC, EnsLD, EnsCT, EnsPseAAC). EnAm-Con first concatenates four feature representations and then integrated 12 trained DNNs in the same way as EnAmDNN. For EnAm-Sep, we separately trained 12-model DNNs based AC, LD, CT, and PseAAC, and integrated these DNNs in the same way as EnAmDNN. The performance of EnAm-Con and EnAm-Sep which emphasized in bold are also listed in **Table 7** where it can be observed that the LD method performs better than AC and PseAAC method. The LD method of AUC value obtained from the first four data sets are 6.2, 4.89, 3.57, 7.17, 0.8 and are 16.37, 9.05, 14.19, 17.89, 9.46% higher than AC and PseAAC methods separately, which is because LD can encode more interaction information. It can be seen from **Table 6** that EnAm-Con performs better than EnAm-Sep, proving that concatenating different feature representations as new feature vector can improve the accuracy of the ensemble strategy. It can be concluded that these four representations are complementary to each other and our ensemble strategy is effective and feasible.

The number of basic learners greatly influences the overall prediction performance, where the efficiency of the model continues to grow as the number of learners increases, to a point that the performance tends to be stable. To evaluate the influence of the EnAmDNN, we assign different numbers of DNNs to protein represent technique, such as 1, 3, 5, 7, 9. The result is presented in **Figure 10**, where it can be observed that the AUC of the EnAmDNN tends to be stable when the number reaches 16. The efficiency of the EnAmDNN may also be affected by the performance of each basic learner, for which we prepare 16 basic learners, iterate them for 600 times, and combine them through deep neural network. The result is shown as follows. It can be seen that the prediction performance improves as the iteration continues and the model tends to remain stable at the point of 200.

**Table 8** reports the process running time of EnAmDNN based on fold cross-validation with 16 basic learners and iterate each basic learner for 600 times.

Meanwhile, to further investigate the contribution of using an ensemble predictor with fold cross-validation, we integrated the simplified EnAmDNN, which don't use fold cross-validation. To reduce the impact of data dependency in the experiment, we constructed data sets on Cardiac based on Muley and Ranjan (2012) to observe the performance of proposed model. From **Table 9**, we can see that EnAmDNN achieves competent prediction performance with an average accuracy of 85.66%, precision of 89.47%, $F_1$ of 85.16%, and AUC of 85.76. It has better performance than simplified EnAmDNN across evaluation metrics. The prediction results show that EnAmDNN with fold cross-validation is capable of predicting PPIs.

## CONCLUSIONS

In this paper, we propose an ensemble deep learning framework (EnAmDNN) with an attention mechanism that aims to predict protein interaction networks. EnAmDNN firstly extracts the

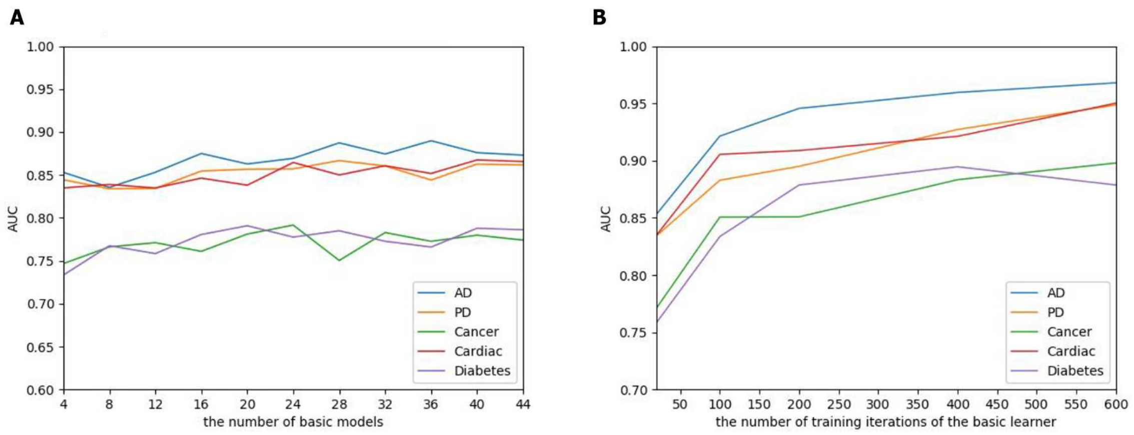

**FIGURE 10 | (A)** The influence of the number of basic learners on the EnAmDNN. It can be observed that the AUC of the EnAmDNN tends to be stable when the number reaches 16. **(B)** The influence of the number of training iterations of the basic learners on the EnAmDNN. It can be seen that the model tends to remain stable at the point of 200.

**TABLE 8 |** Running time of EnAmDNN based on fold cross-validation.

| Date sets | AD | PD | Cancer | Cardiac | Diabetes |
|---|---|---|---|---|---|
| Time (s) | 1,273,589.80 | 835,292.29 | 2,618,297.11 | 1,145,029.32 | 101,784.60 |

**TABLE 9 |** Comparison of EnAmDNN and simplified EnAmDNN.

| Models | Accuracy | Recall | Precision | F1 | AUC |
|---|---|---|---|---|---|
| EnAmDNN | 0.8749 | 0.8138 | 0.9337 | 0.8688 | 0.8765 |
| Simplified EnAmDNN | 0.8566 | 0.8142 | 0.8947 | 0.8516 | 0.8576 |

feature information of protein sequences through AC, LD, CT, and PseAAC, and projects the information into various feature spaces to segment information of AC, LD, CT, PseAAC amino acid from different perspective; then the multi-head attention mechanism is adopted to capture the internal connections of interactions; each technique is assigned 4 independent DNNs with different configurations, resulting in 16 basic learners, and finally combined by deep neural network. To further evaluate the prediction performance of EnAmDNN, we apply it to 5 independent data sets, where improvements of various degrees can be observed for indicators AUC, ACC, Recall, Precision, F1, from which it can be concluded that EnAmDNN learns better than previous approaches from different DNNs and representations.

## DATA AVAILABILITY STATEMENT

Alzheimer disease data was downloaded from the IntAct database (https://www.ebi.ac.uk/intact/query/annot:%22dataset:alzheimers%22?conversationContext=6) under search term annot: "dataset:alzheimers." Cardiac data was downloaded from the IntAct database (https://www.ebi.ac.uk/intact/query/annot:%22dataset:cardiac%22?conversationContext=7) under search term annot: 'dataset:cardiac." Diabetes data was downloaded from the IntAct database (https://www.ebi.ac.uk/intact/query/annot:%22dataset:diabetes%22?conversationContext=8) under search term annot: "dataset:diabetes." Parkinson's disease data was downloaded from the IntAct database (https://www.ebi.ac.uk/intact/query/annot:%22dataset:parkinsons%22?conversationContext=9) under search term annot: "dataset:parkinsons." Cancer disease data was downloaded from the IntAct database (https://www.ebi.ac.uk/intact/query/annot:%22dataset:cancer%22?conversationContext=b) under search term annot: "dataset:cancer." Protein sequence information were downloaded from the Uniprot database (https://www.uniprot.org/downloads).

## AUTHOR CONTRIBUTIONS

Conceived and designed the experiments, performed the experiments, and contributed reagents, materials, analysis tools: FL. Algorithm design and analysis and analyzed the data: FZ and FL. Wrote the paper: FL, FZ, XL, and QL.

## FUNDING

This work was supported by National Natural Science Foundation of China (61303108); The Natural Science Foundation of Jiangsu Higher Education Institutions of China (17KJA520004); Suzhou Key Industries Technological Innovation-Prospective Applied Research Project (SYG201804); Program of the Provincial Key Laboratory for Computer Information Processing Technology (Soochow University) (KJS1524); A Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

## REFERENCES

Alipanahi, B., Delong, A., Weirauch, M. T., and Frey, B. J. (2015). Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol*. 33, 831–838. doi: 10.1038/nbt.3300

Bairoch, A., Apweiler, R., Wu, C. H., Barker, W. C., Boeckmann, B., Ferro, S., et al. (2004). The universal protein resource (UniProt). *Nucleic Acids Res*. 33, 154–159. doi: 10.1093/nar/gki070

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell*. 35, 1798–1828. doi: 10.1109/TPAMI.2013.50

Boxem, M., Maliga, Z., Klitgord, N., Li, N., Lemmens, I., Mana, M., et al. (2008). A protein domain-based interactome network for C. Elegans early embryogenesis. *Cell* 134, 534–545. doi: 10.1016/j.cell.2008.07.009

Chen, Q., Zhu, X., Ling, H.-Z., Wei, S., Jiang, H., and Inkpen, D. (2017). Recurrent neural network-based sentence encoder with gated attention for natural language inference. *arXiv* arXiv:1708.01353. Available online at: https://arxiv.org/abs/1705.00106

Du, X., Shao, J., and Cardie, C. (2017a). Learning to ask: neural question generation for reading comprehension. *arXiv: Comput. Lang*. arXiv:1705.00106v7. Available online at: https://arxiv.org/abs/1708.01353

Du, X., Sun, S., Hu, C., Yao, Y., Yan, Y., Zhang, Y. (2017b). DeepPPI: boosting prediction of protein–protein interactions with deep neural networks. *J. Chem. Inf. Model* 57, 1499–1510. doi: 10.1021/acs.jcim.7b00028

Gomes, H. M., Barddal, J. P., Enembreck, F., and Bifet, A. (2017). A survey on ensemble learning for data stream classification. *ACM Comput. Surv*. 50:23. doi: 10.1145/3054925

Guo, Y., Yu, L., Wen, Z., and Li, M. (2008). Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. *Nucleic Acids Res*. 36, 3025–3030. doi: 10.1093/nar/gkn159

Han, F., and Huang, D. S. (2006). Improved extreme learning machine for function approximation by encoding a priori information. *Neurocomputing* 69, 2369–2373. doi: 10.1016/j.neucom.2006.02.013

Hashemifar, S., Neyshabur, B., Khan, A. A., and Xu, J. (2018). Predicting protein–protein interactions through sequence-based deep learning. *Bioinformatics* 34, 802–810. doi: 10.1093/bioinformatics/bty573

Huang, D. S., and Du, J. X. (2008). A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks. *IEEE Trans. Neural Netw*. 19, 2099–2115. doi: 10.1109/TNN.2008.2004370

Huang, D. S., and Zheng, C. H. (2006). Independent component analysis-based penalized discriminant method for tumor classification using gene expression data. *Bioinformatics* 22, 1855–1862. doi: 10.1093/bioinformatics/btl190

Kerrien, S., Alam-Faruque, Y., Aranda, B., Bancarz, I., Bridge, A., Derow, C., et al. (2007). IntAct—open source resource for molecular interaction data. *Nucleic Acids Res*. 35, 561–565. doi: 10.1093/nar/gkl958

Keskin, O., Tuncbag, N., and Gursoy, A. (2016). Predicting protein–protein interactions from the molecular to the proteome level. *Chem. Rev*. 116, 4884–4909. doi: 10.1021/acs.chemrev.5b00683

Krawczyk, B., Minku, L. L., Gama, J., Stefanowski, J., and Wozniak, M. (2017). Ensemble learning for data stream analysis. *Inform Fus*. 37, 132–156. doi: 10.1016/j.inffus.2017.02.004

Leung, M. K., Xiong, H. Y., Lee, L. J., and Frey, B. J. (2014). Deep learning of the tissue-regulated splicing code. *Bioinformatics* 30, 121–129. doi: 10.1093/bioinformatics/btu277

Li, J., Zhu, X., and Chen, J. Y. (2009). Building disease-specific drug-protein connectivity maps from molecular interaction networks and pubmed abstracts. *PLoS Comput. Biol*. 5:e1000450. doi: 10.1371/journal.pcbi.1000450

Liu, F., Ren, C., Li, H., Zhou, P., Bo, X., and Shu, W. (2016). *De novo* identification of replication-timing domains in the human genome by deep learning. *Bioinformatics* 32, 641–649. doi: 10.1093/bioinformatics/btv643

Liu, S., Shen, F., Komandur Elayavilli, R., Wang, Y., Rastegar-Mojarad, M., Chaudhary, V., et al. (2018). Extracting chemical–protein relations using attention-based neural networks. *Database* 2018:bay102. doi: 10.1093/database/bay102

Martin, S., Roe, D., and Faulon, J. L. (2005). Predicting protein–protein interactions using signature products. *Bioinformatics* 21, 218–226. doi: 10.1093/bioinformatics/bth483

Muley, V. Y., and Ranjan, A. (2012). Effect of reference genome selection on the performance of computational methods for genome-wide protein-protein interaction prediction. *PLoS ONE* 7:e42057. doi: 10.1371/journal.pone.0042057

Planas-Iglesias, J., Bonet, J., García-García, J., Marín-López, M. A., Feliu, E., and Oliva, B. (2013). Understanding protein–protein interactions using local structural features. *J. Mol. Biol*. 425, 1210–1224. doi: 10.1016/j.jmb.2013.01.014

Shen, J., Zhang, J., Luo, X., Zhu, W., Yu, K., Chen, K., et al. (2007). Predicting protein–protein interactions based only on sequences information. *Proc. Natl. Acad. Sci. U. S. A*. 104, 4337–4341. doi: 10.1073/pnas.0607879104

Sun, T., Zhou, B., Lai, L., and Pei, J. (2017). Sequence-based prediction of protein protein interaction using a deep-learning algorithm. *BMC Bioinform*. 18, 277. doi: 10.1186/s12859-017-1700-2

Tian, B., Wu, X., Chen, C., Qiu, W., Ma, Q., and Yu, B. (2019). Predicting protein-protein interactions by fusing various chou's pseudo components and using wavelet denoising approach. *J. Theor. Biol*. 462, 329–346. doi: 10.1016/j.jtbi.2018.11.011

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). *Attention is all You Need. Neural Information Processing Systems (NIPS 2017)*. Long Beach, CA: MIT Press.

Verga, P., Strubell, E., and Mccallum, A. (2018). Simultaneously self-attending to all mentions for full-abstract biological relation extraction. *N Am. Chapter Assoc. Comput. Linguist*. 1, 872–884. doi: 10.18653/v1/N18-1080

Wang, D., Zeng, S., Xu, C., Qiu, W., Liang, Y., Joshi, T., et al. (2017). MusiteDeep: a deep-learning framework for general and kinase-specific phosphorylation site prediction. *Bioinformatics* 33, 3909–3916. doi: 10.1093/bioinformatics/btx496

Wang, X., Yu, B., Ma, A., Chen, C., Liu, B., and Ma, Q. (2019). Protein–protein interaction sites prediction by ensemble random forests with synthetic minority oversampling technique. *Bioinformatics* 35, 2395–2402. doi: 10.1093/bioinformatics/bty995

Xuan, P., Cao, Y., Zhang, T., Kong, R., and Zhang, Z. (2019). Dual convolutional neural networks with attention mechanisms based method for predicting disease-related lncRNA genes. *Front. Genet*. 10:416. doi: 10.3389/fgene.2019.00416

Yang, L., Xia, J. F., and Gui, J. (2010). Prediction of protein-protein interactions from protein sequence using local descriptors. *Protein Pept. Lett*. 17, 1085–1090. doi: 10.2174/092986610791760306

You, W., Yang, Z., Guo, G., Wan, X., and Ji, G. (2019). Prediction of DNA-binding proteins by interaction fusion feature representation and selective ensemble. *Knowledge Based Syst*. 163, 598–610. doi: 10.1016/j.knosys.2018.09.023

Zhang, L., Yu, G., Xia, D., and Wang, J. (2019a). Protein–protein interactions prediction based on ensemble deep neural networks. *Neurocomputing* 324, 10–19. doi: 10.1016/j.neucom.2018.02.097

Zhang, Q. C., Petrey, D., Deng, L., Qiang, L., Shi, Y., Thu, C. A., et al. (2012). Structure-based prediction of protein–protein interactions on a genome-wide scale. *Nature* 490, 556–560. doi: 10.1038/nature11503

Zhang, Y., Lin, H., Yang, Z., Wang, J., and Sun, Y. (2019b). Chemical–protein interaction extraction via contextualized word representations and multihead attention. *Database* 2019:baz054. doi: 10.1093/database/baz054

Zhou, J., and Troyanskaya, O. G. (2015). Predicting effects of noncoding variants with deep learning-based sequence model. *Nat. Methods* 12, 931–934. doi: 10.1038/nmeth.3547

Zhou, Y. Z., Gao, Y., and Zheng, Y. Y. (2011). Prediction of protein-protein interactions using local description of amino acid sequence. *Comm. Comput. Inf*. 202, 254–262. doi: 10.1007/978-3-642-22456-0_37

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