JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

pubs.acs.org/jcim

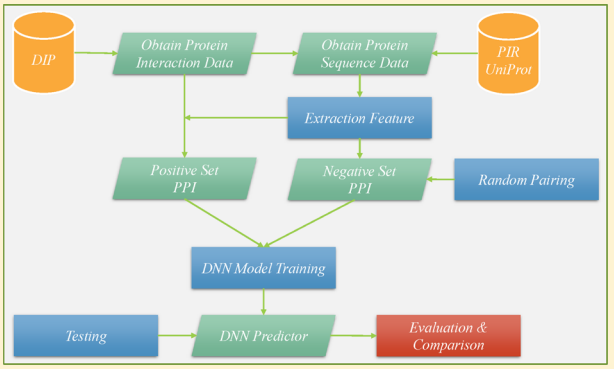
DeepPPI: Boosting Prediction of Protein−Protein Interactions with Deep Neural Networks

Xiuquan Du,*[,†,‡,§] Shiwei Sun,[‡] Changlin Hu,[‡] Yu Yao,[‡] Yuanting Yan,[†,‡,§] and Yanping Zhang[†,‡,§]

[†]Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, [‡]School of Computer Science and Technology, and [§]Center of Information Support & Assurance Technology, Anhui University, Hefei, 230601 Anhui, China

Supporting Information

ABSTRACT: The complex language of eukaryotic gene expression remains incompletely understood. Despite the importance suggested by many proteins variants statistically associated with human disease, nearly all such variants have unknown mechanisms, for example, protein−protein interactions (PPIs). In this study, we address this challenge using a recent machine learning advance-deep neural networks (DNNs). We aim at improving the performance of PPIs prediction and propose a method called DeepPPI (Deep neural networks for Protein−Protein Interactions prediction), which employs deep neural networks to learn effectively the representations of proteins from common protein descriptors. The experimental results indicate that DeepPPI achieves superior performance on the test data set with an Accuracy of 92.50%, Precision of 94.38%, Recall of 90.56%, Specificity of 94.49%, Matthews Correlation Coefficient of 85.08% and Area Under the Curve of 97.43%, respectively. Extensive experiments show that DeepPPI can learn useful features of proteins pairs by a layer-wise abstraction, and thus achieves better prediction performance than existing methods. The source code of our approach can be available via http://ailab.ahu.edu.cn:8087/DeepPPI/index.html.



1. INTRODUCTION

Many critical functions and processes in biology are sustained largely by different types of protein−protein interactions (PPIs), and their misregulation has been associated with many human diseases. Hence the prediction of PPIs is crucial for the understanding of biological processes and various experimental methods targeted to the identification of new PPIs are also proposed. However, the amount of genomic information continues to grow exponentially, the functional annotation of both proteins and their interactions is updated at a slower speed. Conventionally, some methods are used to detect protein interactions both in vitro and in vivo, such as Tandem Affinity Purification (TAP),[1] affinity chromatography,[2,3] Co-Immunoprecipitation (Co-IP),[4] X-ray crystallography,[5] Nuclear Magnetic Resonance (NMR)[6] and the Yeast Two-Hybrid system (Y2H).[7] These experimental techniques have contributed to the generation of databases containing large data sets of protein−protein interaction pairs, such as the Database of Interacting Proteins (DIPs),[8] the Mammalian Protein−Protein Interaction Database (MIPS),[9] the Biomolecular Interaction Network Database (BIND),[10] the IntAct molecular interaction database (IntAct)[11] and the Molecular INTeraction database (MINT)[12] etc. However, genome-scale experiments are costly and labor-intensive, and have inherent biases and limited coverage. Limitations of equipment resolution and environmental disturbances during operations (such as purification, capture, equilibrium, signal label and imaging) could inevitably lead to errors and biases in experimental techniques.[13,14]

Because of shortcomings of experimental methods as mentioned above, computational methodologies have been explored to predict PPIs. The initial strategies include comparative analysis such as the phylogenetic profiling of fused homologues into a single chain obtained from different organisms[15] or other gene fusion methods such as Rosetta Stone[16] mentioned. The conserved gene neighborhood analysis of nine bacterial and archaeal genomes has been performed by Dandekar et al.[17] and proved that the products of conserved genes are likely to interact. Wuchty et al.[18] proposed the domain co-occurrence scale free interaction network. These methods rely on information about protein functional domains, genes and functional pathways among the related species. In addition, proteins' physicochemical properties also be used to generate statistical models and train machine learning algorithms. Bock et al.[19] successfully trained the support vector machine (SVM) using both primary structure information and physicochemical properties of proteins with the Database of Interacting Proteins (DIPs) data set. Gomez et al.[20] described an attraction repulsion model, in which the interaction between protein pair was represented as the sum of the attractive and repulsive forces

Received: January 15, 2017
Published: May 17, 2017

DOI: 10.1021/acs.jcim.7b00028
J. Chem. Inf. Model. 2017, 57, 1499−1510

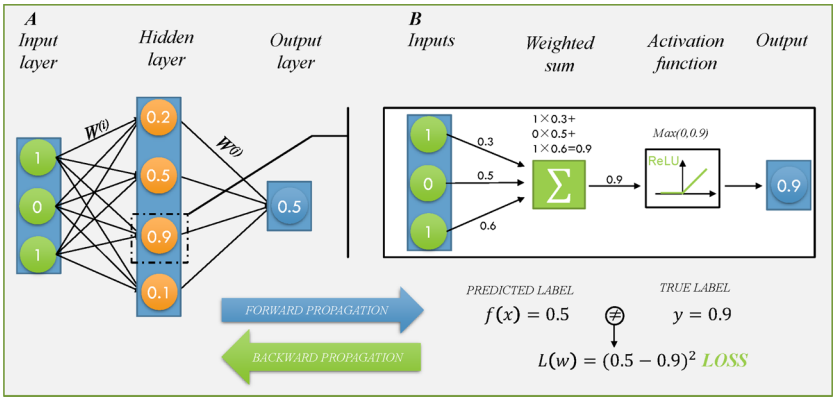


Figure 1. Neural network training procedure.

associated with the domain or motif-sized features. There have also other machine learning model including SVMs,[21−23] Neural Network (NN),[24] Naive Bayes,[25,26] K-Nearest Neighbors,[27] Decision Tree[28,29] and Random Forest[30,31] have been used to predict PPIs. Despite the popularity of PPI prediction methods, there are still some limitations.

First of all, in the above studies, their extracted features for proteins are hand-crafted. Hand-crafting discriminant features or rules for protein requires strong domain knowledge. However, how to select the features plays a crucial role in machine learning models. Second, previous studies mainly extracted information from observed sequences,[28,30] but they generally get lowly discriminant features because of feature noises in the observed sequences, and general machine learning models might not well handle hidden associations from the noise inputs. Third, even more important for machine learning models, it is indispensable to mine refined features buried in noise inputs via multiple abstractions and refinements. Therefore, if one can automatically extract high-level discriminant features from some common protein descriptors, then the proposed method will be expected to be more robust in real-world applications.

Deep neural networks provide a powerful solution for this kind of problem; it consists of model architectures with multiple layers of neural networks,[32−34] which can extract high-level abstractions from data automatically. Meanwhile, deep neural networks has shown better performance than other popular machine learning methods in some research areas, such as speech recognition,[35] signal recognition[33] etc. It also has been proved to be powerful in bioinformatics.[36−38] For example, deep neural networks has been successfully applied to predict RNA splicing patterns in and across various tissues.[38] DeepBind applied deep neural networks to determine sequence specificities of DNA and RNA binding protein, which outperforms other state-of-the-art methods.[37] Similarly, DeepSEA learned regulatory sequence code from chromatin-profiling sequences using deep neural networks, to prioritize further functional variants.[36] Other successful examples of applying deep learning techniques include genomic information extraction,[39−41] protein structure prediction[42] as well as drug discovery.[43] In summary, deep neural networks has the following advantages compared with other sequence-based methods: (1) It can automatically learn specific sequence motifs for RNA-protein.[37] (2) It is able to reduce the impact of noises in the original data and learn real hidden high-level features.[33] Furthermore, some deep-neural-networks-based methods even artificially introduce noises to reduce overfitting, which can enhance model generalization and robustness.[44] In this study, we are inspired by the deep neural network and propose a method called DeepPPI, which employs deep neural networks (DNNs) to learn effectively the representations of proteins pairs. Our contributions are summarized as follows:

(1) The newly designed network architectures can automatically extract abstraction features from sequence features of proteins, and is able to learn sequence specificities for proteins.
(2) We applied deep neural networks to fuse better the learned high-level features from raw input features of proteins.
(3) We use two separate networks as input so that the neural networks can better learn the characteristics of each protein, rather than directly connecting the two proteins

2. MATERIALS AND METHODS

2.1. Data Set Construction. To develop a PPI prediction model, we need to construct or select a valid benchmark data set to evaluate the predictor. In our experiments, *Saccharomyces cerevisiae* PPIs data set is downloaded from the Database of Interacting Proteins (DIP; version 20160731).[45] This data set contains 22 975 positive pairs. The protein pairs that contain a protein with fewer than 50 amino acids are removed, and then a nonredundant subset is generated with the sequence identity level of 40% by cluster analysis of the CD-HIT program.[46] After these preprocessing procedures, the total positive data set is reduced to 17 257.

Choosing negative examples is a very important for training a predictor of PPIs. The common method is based on annotations of cellular localization.[47] The subcellular location information on the proteins is extracted from Swiss-Prot (http://www.expasy.org/sprot/). 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 negative data are obtained by pairing proteins from one location with proteins from other ones. A maximum of 2500 pairs of proteins is taken at each of the two subcellular locations. The strategies must meet the following requirements:[48,49] (1) the noninteracting pairs cannot appear in the positive data set and (2) the contribution of proteins in the negative set should be as harmonious as possible. Finally, a total of 48 594 negative pairs are generated via this approach. All protein pairs are available in the Supporting Information.

Eight different PPI data sets are used to evaluated the performance of DeepPPI. The first PPI data set, described by You et al.,[50] is collected from the *S. cerevisiae* core subset in the database of interacting proteins (DIP) . The positive and negative data sets are combined into a total of 11 188 protein

DOI: 10.1021/acs.jcim.7b00028
J. Chem. Inf. Model. 2017, 57, 1499−1510

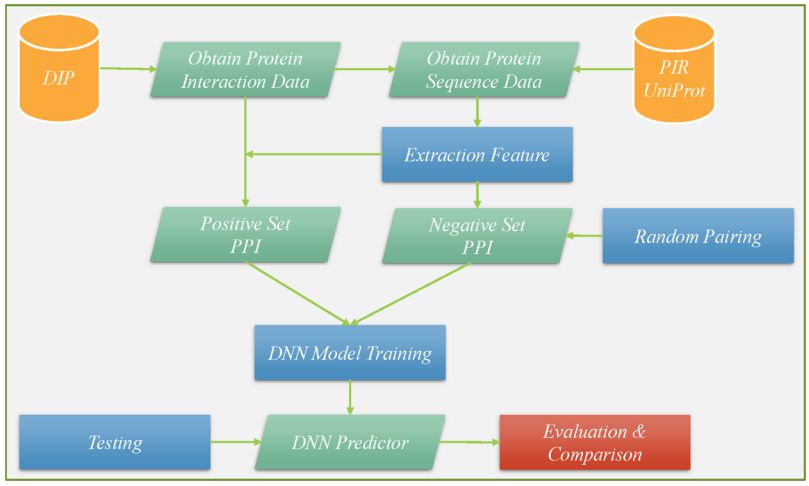


Figure 2. Framework of deep neural networks for protein−protein interactions prediction.

pairs. The second PPI data set, described by Martin et al.,[51] is composed of 2916 *Helicobacter pylori* protein pairs (1458 interacting pairs and 1458 noninteracting pairs). The third PPI data set is collected from Human Protein References Database (HPRD) as described by Huang et al.[52] Huang et al. constructed the Human data set by 8161 protein pairs (3899 interacting pairs and 4262 noninteracting pairs). The other five data sets include *Caenorhabditis elegans* (4013 interacting pairs), *Escherichia coli* (6954 interacting pairs), *Homo sapiens* (1412 interacting pairs), *Mus musculus* (313 interacting pairs), and one additional *H. pylori* data set (1420 interacting pairs) used by Zhou et al.[53] These species-specific PPI data sets are employed in our experiment to verify the effectiveness of our proposed method.

2.2. Deep Neural Network. An artificial neural network, initially inspired by neural networks in the brain,[54−56] consists of layers of interconnected compute units (neurons). The depth of a neural network corresponds to the number of hidden layers, and the width to the maximum number of neurons in one of its layers. As it became possible to train networks with larger numbers of hidden layers, artificial neural networks with a many-layer structure (two or more hidden layers) are called deep neural networks.[34,57]

In the canonical configuration, the network receives data in an input layer, which are then transformed in a nonlinear way through multiple hidden layers, before final outputs are computed in the output layer (Figure 1A). Neurons in a hidden or output layer are connected to all neurons of the previous layer. Each neuron computes a weighted sum of its inputs and applies a nonlinear activation function to calculate its output $f(x)$ (Figure 1B). The most popular activation function is the rectified linear unit (ReLU that thresholds negative signals to 0 and passes through positive signal. This type of activation function allows faster learning compared to alternatives (e.g., sigmoid or tanh unit).[58]

Alternative architectures to such fully connected feed-forward networks have been developed for specific applications, which differ in the way neurons are arranged. These include Convolutional Neural Networks, which are widely used for modeling images, Recurrent Neural Networks for sequential data,[59,60] Restricted Boltzmann Machines[61,62] and Autoencoders[34,63,64] for unsupervised learning. The choice of network architecture and other parameters can be made in a data-driven and objective way by assessing the model performance on a validation data set.

2.3. Design and Implementation of the Prediction System. Figure 2 shows the framework of deep neural networks for protein−protein interactions prediction. This framework consists of the following five steps.

(1) Obtaining protein interaction data from the DIP public database.
(2) According to the protein data from the PIR or UniProt database to retrieve the corresponding protein sequence information.
(3) High quality features of protein sequence are extracted into representative protein.
(4) Negative samples were obtained by random matching of different protein subcellular locations (see details in Section 2.1).
(5) We divide positive sample and negative sample into training set and test set, and feed training set into the deep neural networks.

After, the hyper-parameter adjustment model can be obtained. Finally, we evaluate the prediction performance of the predictor using a set of performance metrics and compare our method with existing prediction approaches.

2.4. Feature Extraction. The deep neural networks model needs a fixed number of inputs for training and testing, thus we need a strategy for encapsulating the global information about proteins of variable length in a fixed length format. The fixed length format is obtained from protein sequences of variable length using sequence-derived structural and physicochemical features, and they are highly useful for representing and distinguishing proteins of different structural, functional and interaction properties, and have been widely used in predicting protein−protein interactions.[19,65,66]

2.4.1. Amino Acid Composition. The amino acid composition is the fraction of each amino acid type within a protein. The amino acid composition gives 20 features and the fractions of all 20 natural amino acids are calculated as

$$fr(r) = \frac{N_r}{N}, \quad r = 1, 2, 3, ..., 20 \tag{1}$$

where N_r is the number of the amino acid type r and N is the length of the sequence.

2.4.2. Dipeptide Composition. The dipeptide composition is used to transform the variable length of proteins to fixed length feature vectors. A dipeptide composition has been used earlier by

DOI: 10.1021/acs.jcim.7b00028
J. Chem. Inf. Model. 2017, 57, 1499−1510

Grassmann et al.[67] and Reczko and Bohr[68] for the development of fold recognition methods. We adopt the same dipeptide composition-based approach in developing a deep neural networks-based method for predicting protein−protein interaction. The dipeptide composition gives a fixed pattern length of 400. Dipeptide composition encapsulates information about the fraction of amino acids as well as their local order. The dipeptide composition is defined as

$$fr(r, s) = \frac{N_{(r,s)}}{N}, \quad r, s = 1, 2, 3, ..., 20 \tag{2}$$

where N the number of dipeptide represented by amino acid type r and s.

2.4.3. Composition, Transition and Distribution. These descriptors are developed by Dubchak et al.[69,70] The amino acids are divided into three classes according to attribute, and each amino acid is encoded by one of the indices 1, 2, 3 according to which class it belongs. The attributes used here include hydrophobicity, normalized van der Waals volume, polarity and polarizability, as in the references. Table S1 shows that amino acid attributes and corresponding division.

For example, given sequence "MTEITAAMVKELRESTGAGA", it will be encoded as "32132223311311222222" according to its hydrophobicity division. The "x" in descriptor ID in Table S1 can be either "1" or "2" or "3", which represents three different feature categories: 1, "Composition (C)"; 2, "Transition (T)"; 3, "Distribution (D)", respectively. Their calculation details for a given attribute are as follows:

2.4.3.1. Composition. It is the global percent for each encoded class in the sequence. In the above example using hydrophobicity division, the numbers for encoded classes "1", "2", "3" are 5, 10, 5 respectively, so the compositions for them are 5/20 = 25%, 10/20 = 50%, 5/20 = 25% respectively, where 20 is the length of the protein sequence. Composition gives 72 features that can be defined as

$$C_r = \frac{N_r}{N}, \quad r = 1, 2, 3 \tag{3}$$

where N_r is the number of r in the encoded sequence and N is the length of the sequence.

2.4.3.2. Transition. A transition from class 1 to 2 is the percent frequency with which 1 is followed by 2 or 2 is followed by 1 in the encoded sequence. Transition descriptor gives 72 features that can be calculated as

$$T_{rs} = \frac{N_{rs} + N_{sr}}{N}, \quad rs = 12, 13, 23 \tag{4}$$

where N_{rs}, N_{sr} is the numbers of dipeptide encoded as rs and sr respectively in the sequence and N is the length of the sequence.

2.4.3.3. Distribution. The "distribution" descriptor describes the distribution of each attribute in the sequence. The distribution gives 360 features. There are five "distribution" descriptors for each attribute and they are the position percent in the whole sequence for the first residue, 25% residues, 50% residues, 75% residues and 100% residues respectively, for a specified encoded class. For example, there are 10 residues encoded as "2" in the above example, the positions for the first residue "2", the second residue "2" (25% × 10 = 2), the fifth "2" residue (50% × 10 = 5), the seventh "2" (75% × 10 = 7) and the tenth residue "2" (100% × 10) in the encoded sequence are 2, 5, 15, 17, 20 respectively, so the distribution descriptors for "2" are 10.0 (2/20 × 100), 25.0 (5/20 × 100), 75.0 (15/20 × 100), 85.0 (17/20 × 100), 100.0 (20/20 × 100), respectively.

2.4.4. Quasi-Sequence-Order Descriptors. The sequence-order features can also be used for representing amino acid distribution patterns of a specific physicochemical property along protein or peptide sequence. These descriptors are derived from both the Schneider-Wrede physicochemical distance matrix[71−73] and the Grantham chemical distance matrix[74] between each pair of the 20 amino acids. The d-th rank sequence-order-coupling number is defined as

$$\tau_d = \sum_{i=1}^{N-d} (d_{i,i+d})^2, \quad d = 1, 2, 3, ..., \text{maxlag} \tag{5}$$

where $d_{i,i+d}$ is the distance between the two amino acids at position i and $i + d$. Maxlag is the maximum lag and the length of the protein must be not less than maxlag. The maxlag is equal to 30 in the experiment. For each amino acid type, the type-1 quasi-sequence-order descriptor can be defined as

$$X_r = \frac{f_r}{(\sum_{r=1}^{20} f_r + w \sum_{d=1}^{\text{maxlag}} \tau_d}, \quad r = 1, 2, 3, ..., 20 \tag{6}$$

where is the normalized occurrence of amino acid type-1 and is a weighting factor (w = 0.1). The type-2 quasi-sequence-order is defined as

$$X_d = \frac{w\tau_{d-20}}{(\sum_{r=1}^{20} f_r + w \sum_{d=1}^{\text{maxlag}} \tau_d}, \quad r = 21, 22, 23, ..., 20 + \text{maxlag} \tag{7}$$

In addition to the Schneider-Wrede physicochemical distance matrix used by Chou et al., another chemical distance matrix by Grantham is also used here. The sequence-order features produce a total of (30 + 50) × 2 = 160 descriptors.

2.4.5. Amphiphilic Pseudoamino Acid Composition (APAAC). APAAC (http://www.csbio.sjtu.edu.cn/bioinf/PseAAC/type2.htm) are also called type-2 pseudoamino acid composition. The definitions of these qualities are similar to PAAC descriptors. First, two variables are derived from the original hydrophobicity values $H_1^0(i)$ and hydrophilicity values $H_2^0(i)$ of 20 amino acids (i = 1, 2, ..., 20):[75]

$$H_1(i) = \frac{H_1^0(i) - \sum_{i=1}^{20} \frac{H_1^0(i)}{20}}{\sqrt{\frac{\sum_{i=1}^{20}\left[H_1^0(i) - \sum_{i=1}^{20} \frac{H_1^0(i)}{20}\right]^2}{20}}} \tag{8}$$

$$H_2(i) = \frac{H_2^0(i) - \sum_{i=1}^{20} \frac{H_2^0(i)}{20}}{\sqrt{\frac{\sum_{i=1}^{20}\left[H_2^0(i) - \sum_{i=1}^{20} \frac{H_2^0(i)}{20}\right]^2}{20}}} \tag{9}$$

where $H_1(i)$ and $H_2(i)$, the hydrophobicity and hydrophilicity correlation functions, are defined respectively as

$$H_{i,j}^1 = H_1(i)H_1(j), \quad H_{i,j}^2 = H_2(i)H_2(j) \tag{10}$$

where sequence order factors can be defined as

$$\tau_1 = \frac{1}{N-1} \sum_{i=1}^{N-1} H_{i,i+1}^1, \quad \tau_2 = \frac{1}{N-1} \sum_{i=1}^{N-2} H_{i,i+1}^2 \tag{11}$$

DOI: 10.1021/acs.jcim.7b00028
J. Chem. Inf. Model. 2017, 57, 1499−1510

$$\tau_3 = \frac{1}{N-2}\sum_{i=1}^{N-2} H^1_{i,i+2}, \quad \tau_4 = \frac{1}{N-2}\sum_{i=1}^{N-2} H^2_{i,i+2} \tag{12}$$

$$\tau_{2\lambda-1} = \frac{1}{N-\lambda}\sum_{i=1}^{N-\lambda} H^1_{i,i+\lambda}, \quad \tau_{2\lambda} = \frac{1}{N-\lambda}\sum_{i=1}^{N-\lambda} H^2_{i,i+\lambda} (\lambda < N) \tag{13}$$

then a set of descriptors called "Amphiphilic Pseudo Amino Acid Composition" (APAAC) are defined as

$$p^u = \frac{f_u}{\sum_{i=1}^{20} f_i + w\sum_{j=1}^{2\lambda}\tau_j} (1 < u < 20) \tag{14}$$

$$p^u = \frac{w\tau_u}{\sum_{i=1}^{20} f_i + w\sum_{j=1}^{2\lambda}\tau_j} (20 + 1 < u < 20 + 2\lambda) \tag{15}$$

where w is the weight factor and is taken as $w = 0.5$, and λ is equal to 30 in the experiment. So, we produce a total of $20 + 2 \times 30 = 80$ descriptors.

2.5. Evaluation Criteria. Holdout validation is the standard for deep neural networks. Holdout validation (Figure 3) partitions the full data set randomly into training, validation and test sets. Models are trained with different hyper-parameters on the training set, from which the model with the highest performance on the validation set is selected. The generalization performance of the model is assessed and compared with other machine learning methods on the test set. The training set is used to learn models with different hyper-parameters, which are then assessed on the validation set. The model with best performance, such as prediction accuracy or mean-squared error, is selected and further evaluated on the test set to quantify the performance on unseen data and for comparison to other methods. In our experiments, data set proportions are 75% for training, and 25% for model testing. We also use 77% of the data to train and 33% to evaluate the model in training set.

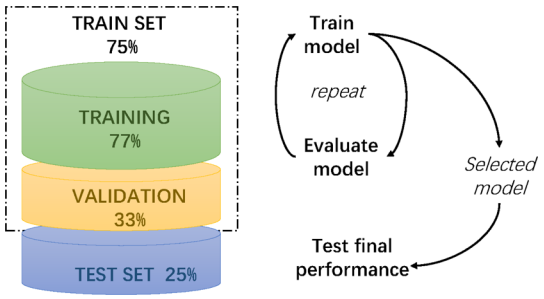


Figure 3. Holdout validation.

The following assessments are taken into account to perform evaluation: Overall Prediction Accuracy, Recall, Specificity, Precision, Matthews Correlation Coefficient (MCC), F_1 score values, Receiver Operating Characteristic (ROC) and Area Under the ROC Curve (AUC). These assessments compute the accuracy and deviation to evaluate the feasibility and robustness of a PPI prediction method. Some are defined as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{16}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{17}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{18}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{19}$$

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \tag{20}$$

$$F_1 = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{21}$$

TP (true positive) is the number of the predicted PPIs found in the positive data set; FP (false positive) is the number of the predicted PPIs not found in positive data set; FN (false negative) is the number of PPIs in the negative data set that failed to be predicted by the method false positive; TN (true negative) is the number of true noninteracting pairs predicted correctly. MCC is a measure of the quality of binary classification, which is a correlation coefficient between the observed and predicted results (it returns a value between −1 and +1. MCC equal to 0 is regarded as a completely random prediction, −1 is regarded as a completely wrong prediction and 1 is regarded as a perfect

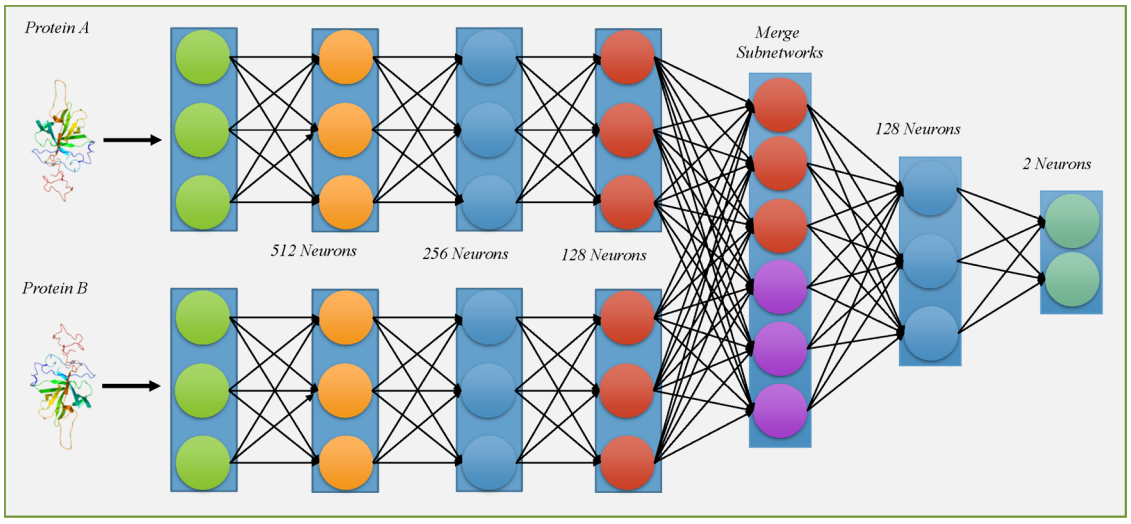


Figure 4. DeepPPI-Sep network architectures.

DOI: 10.1021/acs.jcim.7b00028
J. Chem. Inf. Model. 2017, 57, 1499−1510

Table 1. Comparison among Different Layer Architectures for DeepPPI on *S. cerevisiae*

Architectures	Data Set	Accuracy	Precision	Recall	Specificity	MCC	AUC
DeepPPI-Sep	data set 0	92.70%	94.43%	90.92%	94.51%	85.46%	97.52%
	data set 1	92.35%	94.41%	90.21%	94.54%	84.79%	97.46%
	data set 2	92.75%	94.35%	91.11%	94.42%	85.55%	97.49%
	data set 3	92.10%	94.11%	90.01%	94.23%	84.28%	97.35%
	data set 4	92.62%	94.61%	90.56%	94.73%	85.32%	97.34%
	mean	92.50%	94.38%	90.56%	94.49%	85.08%	97.43%
DeepPPI-Con	data set 0	90.14%	91.57%	88.65%	91.65%	80.32%	95.83%
	data set 1	89.96%	91.66%	88.17%	91.80%	80.00%	95.84%
	data set 2	90.05%	91.19%	88.93%	91.21%	80.14%	95.82%
	data set 3	89.88%	91.47%	88.22%	91.58%	79.82%	95.68%
	data set 4	90.04%	91.63%	88.37%	91.75%	80.14%	95.64%
	mean	90.01%	91.50%	88.47%	91.60%	80.08%	95.76%

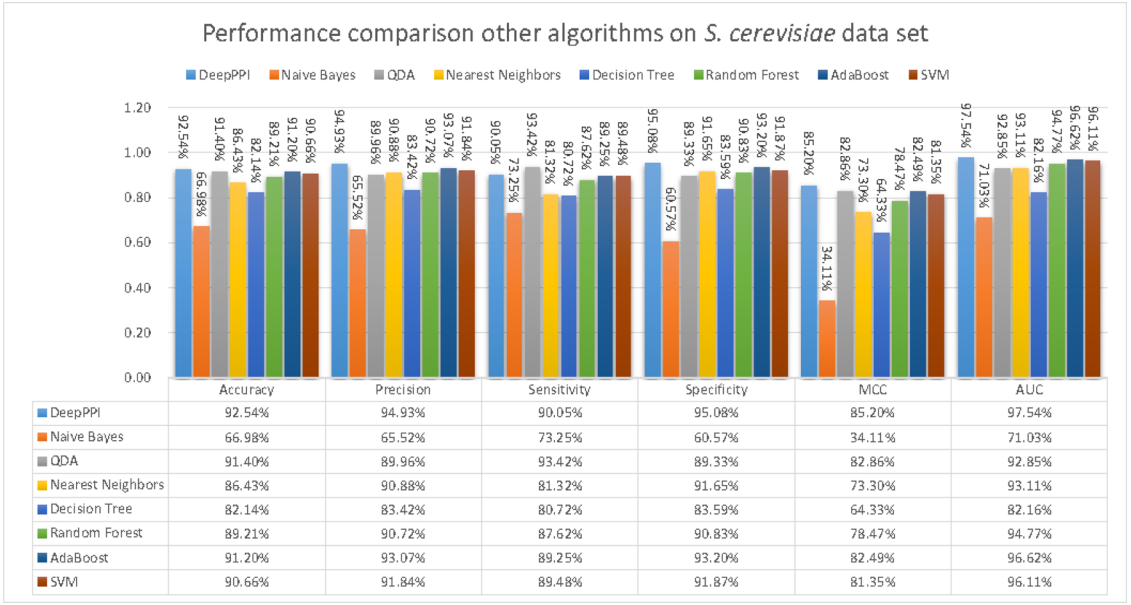


Figure 5. Performance comparison other algorithms on *S. cerevisiae* data set.

prediction). In statistical analysis of binary classification, the F_1 score (also F-score or F-measure) is a measure of test accuracy. It considers both the precision and the recall of the test to compute the score. The F_1 score can be interpreted as a weighted average of the precision and recall, where an F_1 score reaches its best value at 1 and worst at 0. In addition, ROC curve and AUC value illustrate performance of a binary classifier system by graphical plot. The ROC curve is generated by plotting the TP rate against the FP rate at various thresholds, and AUC values are used for comparison between methods.

3. RESULTS

3.1. Performance of PPI Prediction. The proposed predictor is applied to the *S. cerevisiae* data set. The data set consists of 17 257 positive pairs and 48 594 negative pairs. First, we use a 1:1 ratio of positive and negative samples as our data set, which 17 257 negative samples are randomly selected from 48 594 negative pairs. Then holdout validation is used to evaluate DeepPPI. To evaluate the robustness of DeepPPI, there is duplication of randomly selected 17 257 samples from 48 594 negative samples and each sample of negative samples is combined with the positive samples as a data set repeated 5 times. Finally, we obtained five data sets (data sets 0−4).

3.1.1. Comparison among Different Layer Architectures for DeepPPI on the S. Cerevisiae. To understand the impact of different network architectures on DeepPPI's performance, we designed two different network architectures: (a) using two separate networks as input for each protein (namely DeepPPI-Sep) and (b) directly connecting the two proteins in a single network (namely DeepPPI-Con).

3.1.1.1. DeepPPI-Sep. It has two separate (Sep) networks (Figure 4). One is for protein A, the other is for protein B; their inputs are common protein descriptors. The hidden layers for two subnetworks are both 512−256−128. Here 512−256−128 means that the numbers of neurons for 3 hidden layers in two separate networks are 512, 256 and 128, respectively. The third layer of the inverse hidden layer is the concatenation of the two subnetworks, and extracts the high dimensional feature of the protein. The penultimate layer is extracted from the common high dimensional features of the two proteins, and containing 128 neurons.

3.1.1.2. DeepPPI-Con. The raw input is concatenation (Con) of sequence features of protein pairs, which connects to one network. The hidden layers for this network are 512−256−128−128, which means 4 hidden layers have 512, 256, 128 and 128 neurons, respectively.

DOI: 10.1021/acs.jcim.7b00028
J. Chem. Inf. Model. 2017, 57, 1499−1510

The last layer uses one-hot encoding label, because it needs to determine the interaction between protein A and protein B. In this paper, one-hot encoding label is adopted, which is to say the protein label encoding into "10" and "01" where "10" represents no interaction and "01" represents interaction. So, our output layer has 2 neurons.

The prediction result of our method on the *S. cerevisiae* data set is shown in Table 1. As shown in Table 1, the performance of DeepPPI-Sep is better than that of DeepPPI-Con. The Accuracy, Precision, Recall, Specificity, MCC and AUC values obtained from DeepPPI-Sep are 2.49%, 2.88%, 2.09%, 2.89%, 5.00% and 1.67% higher than DeepPPI-Con, respectively.

3.1.2. Compared with Traditional Algorithms. To illustrate further the performance of DeepPPI, we compare DeepPPI with several traditional algorithms, including Nearest Neighbors, SVM, Decision Tree, Random Forest, AdaBoost, Naive Bayes, Quadratic Discriminant Analysis (QDA). For *S. cerevisiae* data set, three-fourths of the data sets are used to training, one-fourth are used to evaluate these methods and compute all performance measures. Figure 5 shows the results of various methods on the *S. cerevisiae*.

From Figure 5, it can be observed that a high prediction accuracy of 92.54% is obtained for our proposed model. Prediction accuracy values of other methods are 66.98%, 91.40%, 86.43%, 82.14%, 89.21%, 91.20% and 90.66%, respectively. Our method has the highest prediction accuracy and MCC compared with all of the above methods. As shown in Figure 6, ROC curves are also plotted to compare various methods on the *S. cerevisiae*. The ROC area of DeepPPI is 8% higher than the other approaches mentioned. All of the results prove the validity and feasibility of our predictors and our methods can efficiently improve the prediction accuracy. We use the gridsearch to obtain the optimal parameters of these algorithms; specific parameters and the detailed description can be seen in the Supporting Information. We also show the comparison of training speed. The machine configuration is the Intel (R) Xeon E2520 CPU with 16G of memory. Table 2 shows training speeds among different algorithms. As shown in Table 2, it can be seen that DeepPPI training is faster than Random Forest, AdaBoost and SVM. The Naive Bayes training time is the shortest, but its results are the worst.

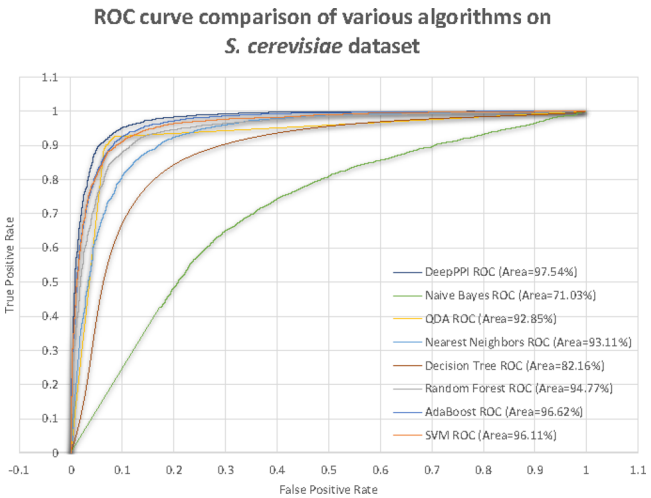


Figure 6. ROC curve comparison of various algorithms on *S. cerevisiae* data set.

3.2. Comparison with Existing Methods. We use eight different PPI data sets to evaluate the performance of our proposed method. The proposed method uses 5-fold cross-validation in comparison to other common approaches on core *S. cerevisiae*, *H. pylori* and *H. sapien* data sets. Then, we test our method on five other data sets, including *H. sapiens*, *M. musculus*, *H. pylori*, *C. elegans* and *E. coli*.

3.2.1. Performance on the S. cerevisiae Core Subset. We compare the prediction performance of our proposed method with that of other existing methods on the *S. cerevisiae* core subset, as shown in Table 3.

We use the same *S. cerevisiae* PPI core subset, and compare our experimental result with those of You et al.,[50,76] Wong et al.,[77] Guo et al.,[48] Zhou et al.[53] and Yang et al.[78] Compared to seven other individual methods, Wong's work achieves the best performance. The average prediction Accuracy, Recall, Precision and MCC are 93.92% ± 0.36%, 91.10% ± 0.31%, 96.45% ± 0.45% and 88.56% ± 0.63%, respectively, which indicate that this method is indeed successful in predicting interactions using a rotation forest model with a novel PR-LPQ descriptor method to describe the information on PPIs. However, the performance of DeepPPI predictor is better than the most accurate predictor. The Accuracy, Recall, Precision and MCC values obtained from the DeepPPI are 0.51%, 0.96%, 0.2% and 0.41% higher than the values of Wong's work, respectively. This result shows the superiority of our method compared to other method.

3.2.2. Performance on the H. Pylori Data Set. DeepPPI also can be tested on the *H. pylori* data set described by Martin et al.[51] In Table 4, we can see that the average prediction performances of our method, such as Accuracy, Recall, Precision and MCC, are 86.23%, 89.44%, 84.32% and 72.63%, respectively. Because *H. pylori* data set is too small, DeepPPI performance is not satisfactory, which we can expect. DeepPPI cannot learn from the data to more useful features; there is no effective feature to represent the proteins.

3.2.3. Performance on H. sapien Data Set. *H. sapien* data set is also used to evaluate DeepPPI, which was used by Huang et al.[52] We compared the prediction performance with Huang's[52] work on *H. sapien* data set, as shown in Table 5. Our method achieves the results with Accuracy, Recall, Precision and MCC are 98.14%, 96.95%, 99.13% and 96.29%, respectively. Our method obtains better prediction results than Huang's work on the *H. sapien* data set, in terms of Accuracy, Recall and MCC.

3.2.4. PPI Identification on Independent Across Species Data Sets. In addition, our methods are tested on five other independent species' data sets. If a large number of physically interacting proteins in one organism exist in a coevolved relationship, their respective orthologs in other organisms interact as well. In this section, we use all 34 514 samples of the *S. cerevisiae* data set as the training set and other species data sets (*E. coli*, *C. elegans*, *H. sapiens*, *H. pylori* and *M. musculus*) as test sets. We use the same feature extraction method as described

Table 2. Comparison of Training Speed among Different Algorithms

Method	DeepPPI	Naive Bayes	QDA	Nearest Neighbors	Decision Tree	Random Forest	AdaBoost	SVM
Time (seconds)	369.12	1.37	34.06	9.25	117.56	3839.22	16961.04	30786.9

DOI: 10.1021/acs.jcim.7b00028
J. Chem. Inf. Model. 2017, 57, 1499−1510

Table 3. Comparison of the Prediction Performance between Our Proposed Method and Other State-of-the-Art Works on the *S. cerevisiae* Core Subset

Method	Accuracy	Recall	Precision	MCC
Our method	94.43% ± 0.30%	92.06% ± 0.36%	96.65% ± 0.59%	88.97% ± 0.62%
You's work[76]	87.00% ± 0.29%	86.15% ± 0.43%	87.59% ± 0.32%	77.36% ± 0.44%
You's work[50]	91.36% ± 0.36%	90.67% ± 0.69%	91.94% ± 0.62%	84.21% ± 0.59%
Wong's work[77]	93.92% ± 0.36%	91.10% ± 0.31%	96.45% ± 0.45%	88.56% ± 0.63%
Guo's work[48]	89.33% ± 2.67%	89.93% ± 3.68%	88.87% ± 6.16%	N/A[a]
Guo's work[48]	87.36% ± 1.38%	87.30% ± 4.68%	87.82% ± 4.33%	N/A
Zhou's work[53]	88.56% ± 0.33%	87.37% ± 0.22%	89.50% ± 0.60%	77.15% ± 0.68%
Yang's work[78]	86.15% ± 1.17%	81.03% ± 1.74%	90.24% ± 1.34%	N/A

[a]Note: N/A means not available.

Table 4. Comparison of the Prediction Performance between Our Proposed Method and Other Methods on the *H. pylori* Data Set

Method	Accuracy	Recall	Precision	MCC
Our method	86.23%	89.44%	84.32%	72.63%
You's work (AC + CT + LD + MAC)[76]	87.50%	88.95%	86.15%	78.13%
You's work (MCD)[50]	84.91%	83.24%	86.12%	74.40%
Huang's work (DCT + SMR)[52]	86.74%	86.43%	87.01%	76.99%
Zhou's work[53]	84.20%	85.10%	83.30%	N/A[a]
Phylogenetic bootstrap[79]	75.80%	69.80%	80.20%	N/A
HKNN[80]	84.00%	86.00%	84.00%	N/A
Signature products[51]	83.40%	79.90%	85.70%	N/A
Ensemble of HKNN[81]	86.60%	86.70%	85.00%	N/A

[a]Note: N/A means not available.

Table 5. Comparison of the Prediction Performance between Our Proposed Method and Other Methods on the *H. sapien* Data Set

Method	Accuracy	Recall	Precision	MCC
Our method	98.14%	96.95%	99.13%	96.29%
Huang's Work(DCT+SMR)[52]	96.30%	92.63%	99.59%	92.82%

above. The performance of these five experiments is summarized in Table 6. The accuracies are 92.19%, 94.84%, 93.77%, 93.66%

Table 6. Prediction Results on Five Independent Species by Our Proposed Method, Based on the *S. cerevisiae* Data Set as the Training Set

	ACC (%)		
Species	Our method	Huang's work[52]	Zhou's work[53]
E.coli	92.19%	66.08%	71.24%
C. elegans	94.84%	81.19%	75.73%
H. sapiens	93.77%	82.22%	76.27%
H. pylori	93.66%	82.18%	N/A[a]
M. musculus	91.37%	79.87%	76.68%

[a]Note: N/A means not available.

and 91.37% on *E. coli*, *C. elegans*, *H. sapiens*, *H. pylori* and *M. musculus* data sets, respectively. It shows that the model is capable of predicting PPIs from other species. The prediction result of our method is better than Huang's work[52] and Zhou's work.[53]

3.3. Comparison with Knowledge-Based PPI Methods. To compare the knowledge-based approach, we used the Gold and Silver data sets provided by Saha et al.;[30] the Yeast and Human Gold data sets included 2117 and 1582 pairs of proteins, respectively. Likewise, there are 14 677 and 27 419 interacting proteins in the corresponding Silver data of the species. In addition to Gold and Silver, in further experiments, the All Interactions data set that contains Human/Yeast PPIs that have been confirmed using at least one experimental method. It contains 57 576 PPIs for human and 190 377 for yeast. For the sake of fairness, we use the same knowledge-based features as Saha et al.[30] The characterization is described as follows: (1) the gene ontology (GO) annotation describes the cellular localization, function and process of a protein; (2) the number of possibly interacting domains in the protein pair; and (3) indicates whether the interaction is supported by the DIPs Paralogous Verification (PVM) method, which only exists in the yeast data set. To be consistent with Saha et al.,[30] we also use 10-fold cross-validation to evaluate performance.

3.3.1. Performance on the Gold and Silver Data Set. DeepPPI performances in Gold and Silver data sets are shown in Table 7 and Table 8. We compare the predictions with those of

Table 7. Average Values of Performance Measures of Different Methods for Yeast Data Set

Data set	Method	Accuracy	Precision	Recall	F_1	AUC
Gold	Our method	0.91	0.92	0.89	0.91	0.97
	Saha-EL[30]	0.91	0.94	0.89	0.91	0.97
	Saha-SVM[30]	0.91	0.93	0.89	0.91	0.97
	Saha-RF[30]	0.89	0.90	0.88	0.89	0.96
	Saha-DT[30]	0.89	0.89	0.88	0.89	0.84
	Saha-NB[30]	0.91	0.94	0.88	0.90	0.97
Silver	Our method	0.79	0.81	0.76	0.78	0.87
	Saha-EL[30]	0.80	0.84	0.73	0.78	0.87
	Saha-SVM[30]	0.79	0.84	0.73	0.78	0.87
	Saha-RF[30]	0.74	0.94	0.51	0.66	0.85
	Saha-DT[30]	0.77	0.80	0.73	0.76	0.77
	Saha-NB[30]	0.78	0.83	0.71	0.77	0.86

Saha et al.[30] on Gold and Silver data sets. Our method on the yeast Gold data set achieves the results with F_1 and AUC of 0.91 and 0.97, respectively. The F_1 and AUC values on the yeast Silver data sets are 0.78 and 0.87 respectively, and the Saha's method basically achieves the same effect. In the human Gold data set, the result of F_1 is 0.89, AUC reaches 0.95. In the human Silver data set, F_1 is 0.80 and AUC reaches 0.87. Compared with Saha's methods, our indicators are not prominent, but the indicators are basically at the same level.

3.3.2. Training in Gold and Silver Data Sets and Testing in All Interactions Data Set. To assess better generalization capability of the trained classifiers, we evaluated them on All Human/Yeast data set. The All Human/Yeast data set is more

DOI: 10.1021/acs.jcim.7b00028
J. Chem. Inf. Model. 2017, 57, 1499−1510

Table 8. Average Values of Performance Measures of Different Methods for Human Data Set

Data set	Method	Accuracy	Precision	Recall	F_1	AUC
Gold	Our method	0.89	0.89	0.89	0.89	0.95
	Saha-EL[30]	0.90	0.89	0.91	0.90	0.95
	Saha-SVM[30]	0.89	0.88	0.91	0.89	0.95
	Saha-RF[30]	0.89	0.92	0.85	0.88	0.95
	Saha-DT[30]	0.87	0.87	0.86	0.86	0.88
	Saha-NB[30]	0.89	0.89	0.90	0.89	0.95
Silver	Our method	0.81	0.82	0.79	0.80	0.87
	Saha-EL[30]	0.81	0.81	0.81	0.81	0.88
	Saha-SVM[30]	0.81	0.80	0.82	0.81	0.86
	Saha-RF[30]	0.80	0.80	0.82	0.81	0.86
	Saha-DT[30]	0.80	0.81	0.78	0.79	0.77
	Saha-NB[30]	0.80	0.81	0.78	0.79	0.85

realistic than the smaller sets. It does not only contain more examples but also introduces class imbalance (there are 3 times more negatives than positives). Classifiers trained on Gold and Silver data set were tested separately. Values of performance measures from this experiment are given in Tables 9 and 10.

Table 9. Values of Performance Measures of Different Methods for Yeast All Data Set

Train data set	Method	Accuracy	Precision	Recall	F_1	AUC
Gold	Our method	0.74	0.47	0.27	0.34	0.61
	Saha-EL[30]	0.64	0.67	0.15	0.25	0.77
	Saha-SVM[30]	0.65	0.66	0.18	0.28	0.77
	Saha-RF[30]	0.68	0.65	0.12	0.21	0.76
	Saha-DT[30]	0.69	1.00	0.23	0.37	0.81
	Saha-NB[30]	0.63	0.70	0.12	0.20	0.77
Silver	Our method	0.70	0.39	0.34	0.36	0.59
	Saha-EL[30]	0.60	0.65	0.13	0.21	0.76
	Saha-SVM[30]	0.60	0.64	0.13	0.21	0.76
	Saha-RF[30]	0.67	0.62	0.19	0.29	0.77
	Saha-DT[30]	0.65	1.00	0.25	0.40	0.81
	Saha-NB[30]	0.61	0.69	0.10	0.18	0.76

Table 10. Values of Performance Measures of Different Methods for Human All Data Set

Train data set	Method	Accuracy	Precision	Recall	F_1	AUC
Gold	Our method	0.85	0.63	0.83	0.71	0.90
	Saha-EL[30]	0.90	0.71	0.61	0.66	0.85
	Saha-SVM[30]	0.89	0.63	0.75	0.68	0.84
	Saha-RF[30]	0.90	0.67	0.73	0.70	0.85
	Saha-DT[30]	0.80	1.00	0.39	0.56	0.86
	Saha-NB[30]	0.88	0.66	0.62	0.64	0.84
Silver	Our method	0.84	0.61	0.86	0.72	0.90
	Saha-EL[30]	0.92	0.74	0.73	0.73	0.88
	Saha-SVM[30]	0.90	0.69	0.74	0.71	0.86
	Saha-RF[30]	0.91	0.73	0.67	0.70	0.87
	Saha-DT[30]	0.84	1.00	0.39	0.56	0.86
	Saha-NB[30]	0.90	0.67	0.75	0.70	0.85

For all yeast, results are far worse, especially in the terms of recall. This means that predictions based on the Gold or Silver subset do not generalize well to the All data set. When interpreting these results, one should remember that some noise is expected in the data set itself. All data set is less reliable than Silver or Gold and may contain much more false positives. Also, among the negative examples there may be some number of interacting protein pairs that have not been discovered yet. For All Human, the results are still relatively good and suggest that the predictors are of practical value. Using the human Gold data set training to predictions in All Human data sets. Compared to Saha-EL, our accuracy and precision metrics are not outstanding. However, the Recall, F_1 and AUC values obtained from our method are 0.22, 0.05 and 0.05 higher than the Saha-EL, respectively using the human Silver data set training to predictions in All Human data sets. Compared with Saha-EL, the accuracy and precision metric are also not satisfactory. But the Recall and AUC values obtained from our method are 0.13 and 0.02 higher than the Saha-EL, respectively.

3.4. Hyper-Parameter Optimization. Table 11 summarizes recommendations and starting points for the most common hyper-parameters. Because the best hyper-parameter configuration is data and application dependent, models with different configurations should be trained and their performance evaluated on a validation set. As the number of configurations and superparameters increase exponentially, trying all of them is impossible in practice.[82] It is therefore recommended to optimize the most important hyper-parameters such as the learning rate and batch size, which is exploring different values while keeping all other hyper-parameters constant. The refined hyper-parameter space can then be further explored by random sampling, and settings with the best performance on the validation set are chosen. Frameworks such as Spearmint,[83] Hyperopt[84] or SMAC[85] allow one to explore automatically the hyper-parameter space using Bayesian optimization. However, although conceptually more powerful, they are at present more difficult to apply and parallelize than random sampling.

Table 11. Central Parameters of a Neural Network and Recommended Settings

Name	Range	Recommendation
Learning rate	1, 0.1, 0.01, 0.001, 0.0001	0.01
Batch size	16, 32, 64, 128, 256	64
Momentum rate	0.8, 0.9, 0.95, 0.99	0.9
Weight initialization	uniform, normal, lecun_uniform, glorot_normal, glorot_uniform, he_normal, he_uniform	glorot_normal
Per-parameter adaptive learning rate methods	SGD, RMSprop, Adagrad, Adadelta, Adam, Adamax, Nadam	SGD
Activation function	relu, tanh, sigmoid, softmax, softplus, softsign, hard_sigmoid	relu
Dropout rate	0.1, 0.2, 0.5, 0.7	0.2

3.5. High-Level Feature Visualization. One merit of the DeepPPI is that it can capture more discriminative features from the raw input data through the hidden layers and thus improve the performance of classification. So, in this subsection, this merit of the DeepPPI is plotted to visualize the features learned from the data. The features are visualized using t-SNE,[86] a technique for high-dimensional data visualization. We visualized the features learned in the first layer and last three hidden layers, which are shown in Figure 7.

From Figure 7, we can see the raw data is very messy. After a merger of the subnetwork layers, there are still some dots of different colors mixing together. With more hidden layers involved in the training, the features become more and more discriminative. After the last hidden layer, the two classes are

DOI: 10.1021/acs.jcim.7b00028
J. Chem. Inf. Model. 2017, 57, 1499−1510

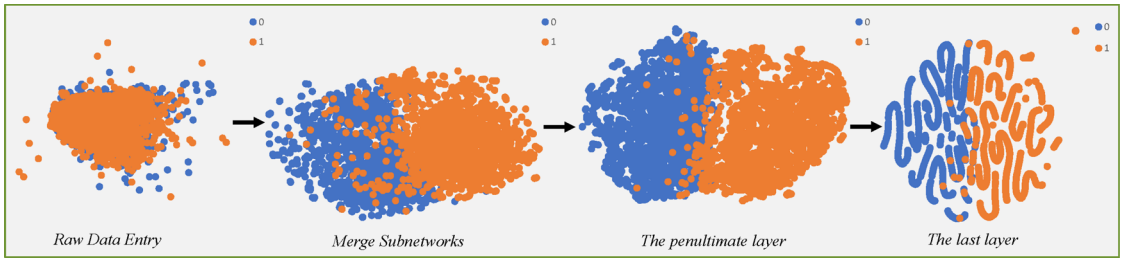


Figure 7. Features learned from merge subnetworks to the last layer hidden layers; each represents the result after one more hidden layer is added, and the two classes are marked by blue (negative) and orange (positive) dots, respectively.

clearly split. This process shows how the DeepPPI learns better with more hidden layers being added. That is, the DeepPPI can extract useful features for prediction by a layer-wise abstraction.

4. DISCUSSION AND CONCLUSION

In this work, our aim is to improve PPI prediction and propose the DeepPPI method that employs the deep learning to extract high-level discriminative features from common protein descriptors. The experimental results show that DeepPPI learns from protein descriptors better than some other ML methods. Concretely, DeepPPI is constructed by a training set of data to obtain a good set of hyper-parameters, then the set of hyper-parameters are applied to test DeepPPI on test sets. Multiple metrics are used to evaluate DeepPPI and compared it with several existing widely used prediction approaches. Experimental results show that DeepPPI generally performs better than the compared approaches.

In this study, one factor that contributes to the good performance of DeepPPI that it can learn useful features of protein pairs by a layer-wise abstraction. Another factor is that DeepPPI can learn an internal distributed feature representation automatically from the data. These characteristics are proved to be an effective approach to learning representations of proteins.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jcim.7b00028.

Amino acid attributes and corresponding division; all proteins pairs; gridsearch to obtain the optimal parameters of ML algorithms (ZIP)

■ AUTHOR INFORMATION

Corresponding Author

*X. Du. E-mail: dxqllp@163.com.

ORCID

Xiuquan Du: 0000-0001-7913-7605

Author Contributions

X.Q.D. conceived the study. X.Q.D., S.W.S. and C.L.H. performed the experiments and analyzed the data. X.Q.D., S.W.S. and Y.Y. drafted the paper. Y.P.Z. and Y.T.Y. provided some suggestions for the paper writing.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work is supported by grants from the National Science Foundation of China (61203290 and 61673020), the Outstanding Young Backbone Teachers Training (02303301), Provincial Natural Science Research Program of Higher Education Institutions of Anhui province (KJ2016A016) and Anhui Provincial Natural Science Foundation (1708085QF143). We also thank the anonymous reviewers for their valuable comments and suggestions, which were helpful for improving the quality of the paper.

■ REFERENCES

(1) Huang, H.; Alvarez, S.; Nusinow, D. A. Data on the Identification of Protein Interactors with the Evening Complex and Pch1 in Arabidopsis Using Tandem Affinity Purification and Mass Spectrometry (Tap-Ms). *Data Brief* **2016**, *8*, 56−60.

(2) Uhlén, M. Affinity as a tool in life science. *BioTechniques* **2008**, *44*, 649−654.

(3) Li, Z. Studies of Drug-Protein Interactions Using High-Performance Affinity Chromatography. Ph.D. Dissertation, University of Nebraska−Lincoln, Lincoln, NB, 2016.

(4) Foltman, M.; Sanchez-Diaz, A. Studying Protein-Protein Interactions in Budding Yeast Using Co-Immunoprecipitation. *Methods Mol. Biol.* **2016**, *1369*, 239−256.

(5) Chou, P. Y.; Fasman, G. D. Prediction of Protein Conformation. *Biochemistry* **1974**, *13*, 222−245.

(6) O'Connell, M. R.; Gamsjaeger, R.; Mackay, J. P. The Structural Analysis of Protein-Protein Interactions by Nmr Spectroscopy. *Proteomics* **2009**, *9*, 5224−5232.

(7) Mehla, J.; Caufield, J. H.; Uetz, P. Mapping Protein-Protein Interactions Using Yeast Two-Hybrid Assays. *Cold Spring Harbor Protoc.* **2015**, *5*, 442−452.

(8) Xenarios, I.; ukasz Salwinski; Duan, X. J.; Higney, P.; Kim, S. M.; A; Eisenberg, D. DIP, the Database of Interacting Proteins: A Research Tool for Studying Cellular Networks of Protein Interactions. *Nucleic Acids Res.* **2002**, *30*, 303−305.

(9) Mewes, H.-W.; Amid, C.; Arnold, R.; Frishman, D.; Güldener, U.; Mannhaupt, G.; Münsterkötter, M.; Pagel, P.; Strack, N.; Stümpflen, V. MIPS: Analysis and Annotation of Proteins from Whole Genomes. *Nucleic Acids Res.* **2004**, *32*, D41−D44.

(10) Bader, G. D.; Hogue, C. W. Bind-A Data Specification for Storing and Describing Biomolecular Interactions, Molecular Complexes and Pathways. *Bioinformatics* **2000**, *16*, 465−477.

(11) Hermjakob, H.; Montecchi-Palazzi, L.; Lewington, C.; Mudali, S.; Kerrien, S.; Orchard, S.; Vingron, M.; Roechert, B.; Roepstorff, P.; Valencia, A. IntAct: An Open Source Molecular Interaction Database. *Nucleic Acids Res.* **2004**, *32*, D452−D455.

(12) Chatraryamontri, A.; Ceol, A.; Palazzi, L. M.; Nardelli, G.; Schneider, M. V.; Castagnoli, L.; Cesareni, G. Mint: The Molecular Interaction Database. *Nucleic Acids Res.* **2007**, *35*, D572−D574.

(13) Piehler, J. New Methodologies for Measuring Protein Interactions in Vivo and in Vitro. *Curr. Opin. Struct. Biol.* **2005**, *15*, 4−14.

(14) Byron, O.; Vestergaard, B. Protein-Protein Interactions: A Supra-Structural Phenomenon Demanding Trans-Disciplinary Biophysical Approaches. *Curr. Opin. Struct. Biol.* **2015**, *35*, 76−86.

(15) Marcotte, E. M.; Pellegrini, M.; Ng, H. L.; Rice, D. W.; Yeates, T. O.; Eisenberg, D. Detecting Protein Function and Protein-Protein Interactions from Genome Sequences. *Science* **1999**, *285*, 751−753.

DOI: 10.1021/acs.jcim.7b00028
J. Chem. Inf. Model. 2017, 57, 1499−1510

(16) Enright, A. J.; Iliopoulos, I.; Kyrpides, N. C.; Ouzounis, C. A. Protein Interaction Maps for Complete Genomes Based on Gene Fusion Events. *Nature* **1999**, *402*, 86−90.
(17) Dandekar, T.; Snel, B.; Huynen, M.; Bork, P. Conservation of Gene Order: A Fingerprint of Proteins That Physically Interact. *Trends Biochem. Sci.* **1998**, *23*, 324−328.
(18) Wuchty, S. Scale-Free Behavior in Protein Domain Networks. *Mol. Biol. Evol.* **2001**, *18*, 1694−1702.
(19) Bock, J. R.; Gough, D. A. Predicting Protein-Protein Interactions from Primary Structure. *Bioinformatics* **2001**, *17*, 455−460.
(20) Gomez, S. M.; Noble, W. S.; Rzhetsky, A. Learning to Predict Protein-Protein Interactions from Protein Sequences. *Bioinformatics* **2003**, *19*, 1875.
(21) Chatterjee, P.; Basu, S.; Kundu, M.; Nasipuri, M.; Plewczynski, D. Ppi_Svm: Prediction of Protein-Protein Interactions Using Machine Learning, Domain-Domain Affinities and Frequency Tables. *Cell Mol. Biol. Lett.* **2011**, *16*, 264−278.
(22) Rashid, M.; Ramasamy, S.; Raghava, G. P. A Simple Approach for Predicting Protein-Protein Interactions. *Curr. Protein Pept. Sci.* **2010**, *11*, 589−600.
(23) Dohkan, S.; Koike, A.; Takagi, T. Improving the Performance of an SVM-Based Method for Predicting Protein-Protein Interactions. *Silico Biol.* **2006**, *6*, 515−529.
(24) Fariselli, P.; Pazos, F.; Valencia, A.; Casadio, R. Prediction of Protein-Protein Interaction Sites in Heterocomplexes with Neural Networks. *Eur. J. Biochem.* **2002**, *269*, 1356−1361.
(25) Lin, X.; Chen, X. W. Heterogeneous Data Integration by Tree-Augmented Naive Bayes for Protein-Protein Interactions Prediction. *Proteomics* **2013**, *13*, 261−268.
(26) Najafabadi, H. S.; Salavati, R. Sequence-Based Prediction of Protein-Protein Interactions by Means of Codon Usage. *Genome Biol.* **2008**, *9*, 1−9.
(27) Browne, F.; Wang, H.; Zheng, H.; Azuaje, F. Supervised Statistical and Machine Learning Approaches to Inferring Pairwise and Module-Based Protein Interaction Networks. *Bioinformatics and Bioengineering, Proceedings of the 7th IEEE International Conference on*; IEEE: New York, 2007; pp 1365−1369.
(28) Valente, G. T.; Acencio, M. L.; Martins, C.; Lemke, N. The Development of a Universal in Silico Predictor of Protein-Protein Interactions. *PLoS One* **2013**, *8*, e65587.
(29) Chen, X. W.; Liu, M. Prediction of Protein-Protein Interactions Using Random Decision Forest Framework. *Bioinformatics* **2005**, *21*, 4394−4400.
(30) Saha, I.; Zubek, J.; Klingström, T.; Forsberg, S.; Wikander, J.; Kierczak, M.; Maulik, U.; Plewczynski, D. Ensemble Learning Prediction of Protein-Protein Interactions Using Proteins Functional Annotations. *Mol. BioSyst.* **2014**, *10*, 820−830.
(31) Qi, Y.; Klein-Seetharaman, J.; Bar-Joseph, Z. Random Forest Similarity for Protein-Protein Interaction Prediction from Multiple Sources. *Pac. Symp. Biocomput.* **2015**, *10*, 531−542.
(32) LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436−444.
(33) Bengio, Y.; Courville, A.; Vincent, P. Representation Learning: A Review and New Perspectives. *IEEE T Pattern Anal* **2013**, *35*, 1798−1828.
(34) Hinton, G. E.; Salakhutdinov, R. R. Reducing the Dimensionality of Data with Neural Networks. *Science* **2006**, *313*, 504−507.
(35) Maas, A. L.; Le, Q. V.; O'Neil, T. M.; Vinyals, O.; Nguyen, P.; Ng, A. Y. Recurrent neural networks for noise reduction in robust ASR. *Interspeech* **2012**, 22−25.
(36) Zhou, J.; Troyanskaya, O. G. Predicting Effects of Noncoding Variants with Deep Learning-Based Sequence Model. *Nat. Methods* **2015**, *12*, 931−934.
(37) Alipanahi, B.; Delong, A.; Weirauch, M. T.; Frey, B. J. Predicting the Sequence Specificities of Dna-and Rna-Binding Proteins by Deep Learning. *Nat. Biotechnol.* **2015**, *33*, 831−838.
(38) Leung, M. K.; Xiong, H. Y.; Lee, L. J.; Frey, B. J. Deep Learning of the Tissue-Regulated Splicing Code. *Bioinformatics* **2014**, *30*, i121−i129.
(39) Zhang, S.; Zhou, J.; Hu, H.; Gong, H.; Chen, L.; Cheng, C.; Zeng, J. A Deep Learning Framework for Modeling Structural Features of RNA-Binding Protein Targets. *Nucleic Acids Res.* **2016**, *44*, e32.
(40) Liu, F.; Ren, C.; Li, H.; Zhou, P.; Bo, X.; Shu, W. De Novo Identification of Replication-Timing Domains in the Human Genome by Deep Learning. *Bioinformatics* **2016**, *26*, 214−216.
(41) Leung, M. K. K.; Delong, A.; Alipanahi, B.; Frey, B. J. Machine Learning in Genomic Medicine: A Review of Computational Problems and Data Sets. *Proc. IEEE* **2016**, *104*, 176−197.
(42) Spencer, M.; Eickholt, J.; Cheng, J. A Deep Learning Network Approach to Ab Initio Protein Secondary Structure Prediction. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2015**, *12*, 103−112.
(43) Gawehn, E.; Hiss, J. A.; Schneider, G. Deep Learning in Drug Discovery. *Mol. Inf.* **2016**, *35*, 3−14.
(44) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. Imagenet Classification with Deep Convolutional Neural Networks. *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS'12)*; Neural Information Processing Systems Foundation, Inc.: La Jolla, CA, 2012; Vol. *25*, pp 1097−1105.
(45) Rédei, G. P. DIP (Database of Interacting Proteins). *Encyclopedia of Genetics Genomics Proteomics & Informatics* **2008**, 511−511.
(46) Li, W.; Godzik, A. Cd-Hit: A Fast Program for Clustering and Comparing Large Sets of Protein or Nucleotide Sequences. *Bioinformatics* **2006**, *22*, 1658−1659.
(47) Shin, C. J.; Wong, S.; Davis, M. J.; Ragan, M. A. Protein-Protein Interaction as a Predictor of Subcellular Location. *BMC Syst. Biol.* **2009**, *3*, 1−20.
(48) Guo, Y.; Yu, L.; Wen, Z.; Li, M. Using Support Vector Machine Combined with Auto Covariance to Predict Protein-Protein Interactions from Protein Sequences. *Nucleic Acids Res.* **2008**, *36*, 3025−3030.
(49) Shen, J.; Zhang, J.; Luo, X.; Zhu, W.; Yu, K.; Chen, K.; Li, Y.; Jiang, H. Predicting Protein-Protein Interactions Based Only on Sequences Information. *Proc. Natl. Acad. Sci. U. S. A.* **2007**, *104*, 4337−4341.
(50) You, Z. H.; Zhu, L.; Zheng, C. H.; Yu, H. J.; Deng, S. P.; Ji, Z. Prediction of Protein-Protein Interactions from Amino Acid Sequences Using a Novel Multi-Scale Continuous and Discontinuous Feature Set. *BMC Bioinf.* **2014**, *15*, S9.
(51) Martin, S.; Roe, D.; Faulon, J. L. Predicting Protein-Protein Interactions Using Signature Products. *Bioinformatics* **2005**, *21*, 218−226.
(52) Huang, Y. A.; You, Z. H.; Gao, X.; Wong, L.; Wang, L. Using Weighted Sparse Representation Model Combined with Discrete Cosine Transformation to Predict Protein-Protein Interactions from Protein Sequence. *BioMed Res. Int.* **2015**, *2015*, 1−10.
(53) Zhou, Y. Z.; Gao, Y.; Zheng, Y. Y. Prediction of Protein-Protein Interactions Using Local Description of Amino Acid Sequence. *Communications in Computer and Information Science* **2011**, *202*, 254−262.
(54) Rosenblatt, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychol. Rev.* **1988**, *65*, 386−408.
(55) Farley, B.; Clark, W. Simulation of Self-Organizing Systems by Digital Computer. *Transactions of the Ire Professional Group on Information Theory* **1954**, *4*, 76−84.
(56) Mcculloch, W. S.; Pitts, W. A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bull. Math. Biophys.* **1943**, *5*, 115−133.
(57) Hinton, G. E.; Osindero, S.; Teh, Y. W. A Fast Learning Algorithm for Deep Belief Nets. *Neural Comput* **2006**, *18*, 1527−1554.
(58) Glorot, X.; Bordes, A.; Bengio, Y. Deep Sparse Rectifier Neural Networks. *J. Mach. Learn Res.* **2011**, *15*, 315−323.
(59) Lipton, Z. C.; Berkowitz, J.; Elkan, C. A Critical Review of Recurrent Neural Networks for Sequence Learning. *Comput. Sci.* **2015**, 1−33.
(60) Sutskever, I. Training Recurrent Neural Networks. Ph.D. Thesis, University of Toronto, Toronto, ON, 2013.
(61) Hinton, G. E. A Practical Guide to Training Restricted Boltzmann Machines. *Neural Networks: Tricks of the Trade* **2012**, *7700*, 599−619.

DOI: 10.1021/acs.jcim.7b00028
J. Chem. Inf. Model. 2017, 57, 1499−1510

(62) Salakhutdinov, R.; Larochelle, H. Efficient Learning of Deep Boltzmann Machines. *AISTATs* **2010**, *9*, 693−700.
(63) Alain, G.; Bengio, Y.; Rifai, S. Regularized Auto-Encoders Estimate Local Statistics. *Proc. CoRR* **2012**, 1−17.
(64) Kingma, D. P.; Welling, M. Auto-Encoding Variational Bayes. *Conference Proceedings: Papers Accepted to the International Conference on Learning Representations* 2014, 1−14.
(65) Qiu, J.; Noble, W. S. Predicting Co-Complexed Protein Pairs from Heterogeneous Data. *PLoS Comput. Biol.* **2008**, *4*, e1000054.
(66) Lo, S. L.; Cai, C. Z.; Chen, Y. Z.; Chung, M. Effect of Training Datasets on Support Vector Machine Prediction of Protein-Protein Interactions. *Proteomics* **2005**, *5*, 876−884.
(67) Grassmann, J.; Reczko, M.; Suhai, S.; Edler, L. Protein Fold Class Prediction: New Methods of Statistical Classification. *ISMB* **1999**, 106−112.
(68) Reczko, M.; Bohr, H. The Def Data Base of Sequence Based Protein Fold Class Predictions. *Nucleic Acids Res.* **1994**, *22*, 3616−3619.
(69) Dubchak, I.; Muchnik, I.; Mayor, C.; Dralyuk, I.; Kim, S.-H. Recognition of A Protein Fold in the Context of the Scop Classification. *Proteins: Struct., Funct., Genet.* **1999**, *35*, 401−407.
(70) Dubchak, I.; Muchnik, I.; Holbrook, S. R.; Kim, S.-H. Prediction of Protein Folding Class Using Global Description of Amino Acid Sequence. *Proc. Natl. Acad. Sci. U. S. A.* **1995**, *92*, 8700−8704.
(71) Schneider, G.; Wrede, P. The Rational Design of Amino Acid Sequences by Artificial Neural Networks and Simulated Molecular Evolution: De Novo Design of an Idealized Leader Peptidase Cleavage Site. *Biophys. J.* **1994**, *66*, 335−344.
(72) Chou, K.-C.; Cai, Y.-D. Prediction of Protein Subcellular Locations by Go-Fund-Pseaa Predictor. *Biochem. Biophys. Res. Commun.* **2004**, *320*, 1236−1239.
(73) Chou, K.-C. Prediction of Protein Subcellular Locations by Incorporating Quasi-Sequence-Order Effect. *Biochem. Biophys. Res. Commun.* **2000**, *278*, 477−483.
(74) Grantham, R. Amino Acid Difference Formula to Help Explain Protein Evolution. *Science* **1974**, *185*, 862−864.
(75) Shen, H. B.; Chou, K. C. Pseaac: A Flexible Web Server for Generating Various Kinds of Protein Pseudo Amino Acid Composition. *Anal. Biochem.* **2008**, *373*, 386−388.
(76) You, Z. H.; Lei, Y. K.; Zhu, L.; Xia, J.; Wang, B. Prediction of Protein-Protein Interactions from Amino Acid Sequences with Ensemble Extreme Learning Machines and Principal Component Analysis. *BMC Bioinf.* **2013**, *14*, 1−11.
(77) Wong, L.; You, Z. H.; Li, S.; Huang, Y. a.; Liu, G. *Detection of Protein−Protein Interactions from Amino Acid Sequences Using A Rotation Forest Model with a Novel PR-LPQ Descriptor*; Springer International Publishing: Cham, Switzerland, 2015; pp 713−720.
(78) Yang, L.; Xia, J. F.; Gui, J. Prediction of Protein-Protein Interactions from Protein Sequence Using Local Descriptors. *Protein Pept. Lett.* **2010**, *17*, 1085−1090.
(79) Bock, J. R.; Gough, D. A. Whole-Proteome Interaction Mining. *Bioinformatics* **2003**, *19*, 125−134.
(80) Nanni, L. Letters: Hyperplanes for Predicting Protein-Protein Interactions. *Neurocomputing* **2005**, *69*, 257−263.
(81) Nanni, L.; Lumini, A. An Ensemble of K-Local Hyperplanes for Predicting Protein-Protein Interactions. *Bioinformatics* **2006**, *22*, 1207−1210.
(82) Bengio, Y. *Practical Recommendations for Gradient-Based Training of Deep Architectures*; Springer: Berlin Heidelberg, 2012; pp 133−144.
(83) Snoek, J.; Larochelle, H.; Adams, R. P. Practical Bayesian Optimization of Machine Learning Algorithms. *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS'12)*; Neural Information Processing Systems Foundation, Inc.: La Jolla, CA, 2012; Vol. *25*, pp 2951−2959.
(84) Bergstra, J.; Cox, D. D. Hyperparameter Optimization and Boosting for Classifying Facial Expressions: How Good Can A "Null" Model Be? *Comput. Sci.* **2013**, 1−6.
(85) Hutter, F.; Hoos, H. H.; Leyton-Brown, K. Sequential Model-Based Optimization for General Algorithm Configuration. *Learning and Intelligent Optimization* **2011**, *6683*, 507−523.
(86) Laurens, V. D. M.; Hinton, G. Visualizing Data Using T-Sne. *J. Mach. Learn Res.* **2008**, *9*, 2579−2605.

DOI: 10.1021/acs.jcim.7b00028
J. Chem. Inf. Model. 2017, 57, 1499−1510