Hindawi Publishing Corporation
BioMed Research International
Volume 2014, Article ID 598129, 9 pages
http://dx.doi.org/10.1155/2014/598129

Research Article

Large-Scale Protein-Protein Interactions Detection by Integrating Big Biosensing Data with Computational Model

Zhu-Hong You,[1] Shuai Li,[2] Xin Gao,[3] Xin Luo,[2] and Zhen Ji[1]

[1] *College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong 518060, China*
[2] *Department of Computing, Hong Kong Polytechnic University, Hong Kong*
[3] *Department of Medical Imaging, Suzhou Institute of Biomedical Engineering and Technology, Suzhou, Jiangsu 215163, China*

Correspondence should be addressed to Shuai Li; shuaili@polyu.edu.hk and Xin Gao; gaox@sibet.ac.cn

Received 23 June 2014; Accepted 24 July 2014; Published 18 August 2014

Academic Editor: Jiangning Song

Protein-protein interactions are the basis of biological functions, and studying these interactions on a molecular level is of crucial importance for understanding the functionality of a living cell. During the past decade, biosensors have emerged as an important tool for the high-throughput identification of proteins and their interactions. However, the high-throughput experimental methods for identifying PPIs are both time-consuming and expensive. On the other hand, high-throughput PPI data are often associated with high false-positive and high false-negative rates. Targeting at these problems, we propose a method for PPI detection by integrating biosensor-based PPI data with a novel computational model. This method was developed based on the algorithm of extreme learning machine combined with a novel representation of protein sequence descriptor. When performed on the large-scale human protein interaction dataset, the proposed method achieved 84.8% prediction accuracy with 84.08% sensitivity at the specificity of 85.53%. We conducted more extensive experiments to compare the proposed method with the state-of-the-art techniques, support vector machine. The achieved results demonstrate that our approach is very promising for detecting new PPIs, and it can be a helpful supplement for biosensor-based PPI data detection.

1. Introduction

Proteins play crucial roles in cellular biology, including signaling cascades, metabolic cycles, and DNA transcription. In most cases, proteins rarely perform their functions alone; instead, they cooperate with other proteins by forming protein-protein interactions (PPIs) networks. PPIs are responsible for the majority of cellular functions. Over the past decades, many innovative techniques and systems for identifying protein interactions have been developed [1]; for example, in the high-throughput experimental technologies such as yeast two-hybrid (Y2H) screens [2], tandem affinity purification (TAP) [3], mass spectrometric protein complex identification (MS-PCI) [4], and other large-scale biological techniques for PPIs detection, a large amount of PPIs data for different species has been accumulated [5–11]. However, the experimental methods are costly and time consuming; therefore, current PPI pairs obtained from biological experiments only cover a small fraction of the complete PPI networks [12–14]. In addition, large-scale experimental methods usually suffer from high rates of both false positives and false negatives [12, 15–20]. Hence, it is of great practical significance to build low cost protein detection systems and establish the reliable computational methods to facilitate the detection of PPIs [21–25].

A number of computational methods have been proposed for the prediction of PPIs based on different data types, including phylogenetic profiles, gene neighborhood, gene fusion, sequence conservation between interacting proteins, and literature mining knowledge [12, 26–33]. There are also methods that combine interaction information from several different data sources [27]. However, the aforementioned methods cannot be carried out if such biological information about the proteins is not available. Recently, a number of methods which derive information directly from protein sequence are of particular interest [26, 28–30]. Researchers are committed to develop the sequences-based method for discovering new PPIs, and the experimental results showed

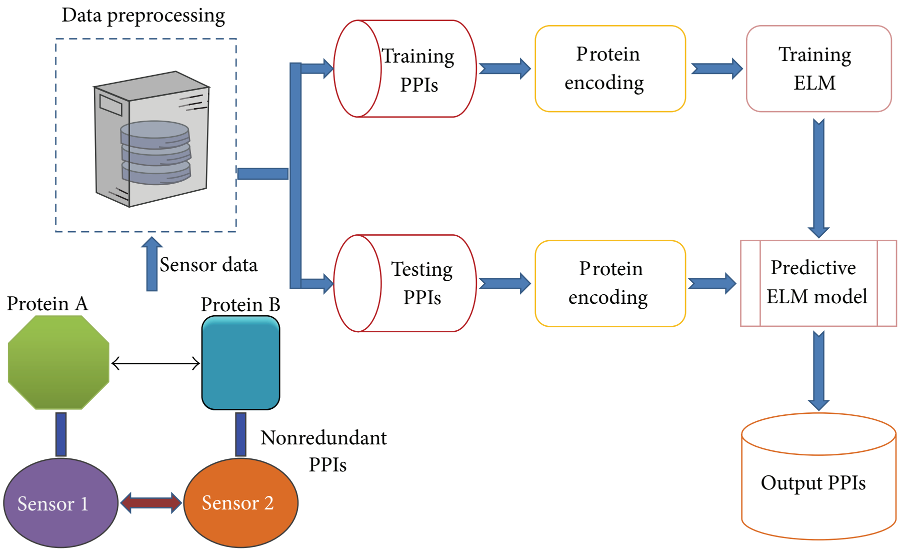

Figure 1: The schematic diagram for mapping large-scale protein-protein interactions by integrating biosensor data with ELM model.

that the information of amino acid sequences of proteins alone is sufficient to predict PPIs. Among them, one of the excellent works is a support vector machine based method developed by Shen et al. [29]. In that study, the twenty amino acids were firstly clustered into 7 classes according to their volumes and dipoles of the side chains. Then the conjoint triad approach extracts the features of protein pairs based on the classification of amino acids. When applied to predict *human* PPIs, this method yields a high prediction accuracy of about 84%.

Because the conjoint triad approach did not take neighboring effect into account and the interactions usually occur in the discontinuous amino acids segments in the sequence, on the other work Guo et al. developed a method based on SVM and autocovariance to extract the interactions information in the discontinuous amino acids segments in the sequence [26]. Their method yielded a prediction accuracy of 86.55%, when applied to predicting *Saccharomyces cerevisiae* PPIs. Lately, Pan et al. proposed a novel hierarchical LDA-RF model to predict *human* PPIs from protein primary sequences directly. In this study, the local sequential features represented by conjoint triads are firstly extracted from sequences. Then the generative LDA model is used to project the original feature space into the latent semantic space to obtain low dimensional latent topic features. Finally, the random forest model is used to predict the interactions between two proteins. The experimental results show that it is a very promising scheme for PPIs prediction [28].

The general trend in the current study for predicting PPIs has focused on high accuracy but has not considered the running time taken to train the classification model, which should be an important factor of developing a sequence-based method for predicting PPIs because the total number of possible PPIs is very large. For example, if we assume that the *human* genome consists of 22,500 protein-coding genes, then the total number of possible PPIs is estimated to be around 253,113,750 ($N = 22,500 \times (22,500 - 1)/2$), which indicates that some classification models with high classification accuracy may not be satisfactory when considering the tradeoff between the classification accuracy and the time for training the models. Here, in addition to exploring the local and global descriptors to mine interaction information from the multiscale amino acids segments at the same time, we also investigate the use of a novel paradigm of learning machine called extreme learning machine (ELM) [34], in order to obtain a balance between high classification accuracy and short training time.

In the present work, we report a novel sequence-based method for the prediction of interacting protein pairs using ELM combined with local and global descriptors. More specifically, we first represent each protein sequence as a vector by utilizing the novel representation of local and global protein sequence descriptors which provides us with a chance to mine interaction information from the multiscale amino acids segments at the same time. Then we characterize a protein pair in different feature vectors by coding the vectors of two proteins in this protein pair. Finally, an ELM model is constructed using these feature vectors of the protein pair as input. To evaluate the performance, the proposed method was applied to *human* PPI dataset. The experiment results show that our method achieved 84.8% prediction accuracy with 84.08% sensitivity at the specificity of 85.53%.

2. Materials and Methodology

In this section, we outline the main idea behind the proposed method. The flowchart intuitively showing how to map large-scale PPIs by integrating biosensor-based PPI data with computational model is given in Figure 1. Firstly, we discuss the PPI dataset which is used in the study to evaluate the performance of the proposed method. Next we introduce the novel sequence-based protein representation method.

Finally, we briefly descript the computational model, ELM, used in this study.

2.1. Golden Standard Datasets. We evaluated the proposed method with the *human* PPI dataset, which was downloaded from the Human Protein References Database (HPRD). After self-interactions and duplicate interactions were removed, the remaining 36,630 PPI pairs between 9,630 different human proteins comprise the final positive dataset.

The chosen golden negative dataset has a variable impact on the prediction performance, and it can be artificially inflated by a bias towards dominant samples in the positive data. For golden negative set, we followed the previous work [28] assuming that the proteins in separate subcellular compartments do not interact with each other. In this study, the golden negative dataset is generated from Swiss-Prot database version 57.3 according to four criteria: (1) protein sequences annotated with uncertain subcellular location terms were removed. (2) Protein sequences annotated by multiple locations were removed because of lack of the uniqueness. (3) Protein sequences annotated with "fragment" were removed. (4) Protein sequences with less than 50 amino acid residues were also removed because they might be fragments. After strictly following the above steps, we finally obtained 1,773 human proteins from six subcellular localizations. Then the noninteracting protein pairs were constructed by randomly pairing the proteins from separate subcellular compartments.

We also downloaded the golden negative dataset of human with experimental evidence used in the study of Smialowski et al. [35]. By combining the above two negative datasets, the whole final golden negative dataset consists of 36,480 noninteracting protein pairs. The whole dataset consists of 73,110 protein pairs, where nearly half are from the positive dataset and half are from the negative dataset. Four-fifths of the protein pairs from the positive and negative dataset were, respectively, randomly selected as the training dataset and the remaining one-fifths were used as the testing dataset.

2.2. Representing Proteins with Descriptors from Primary Protein Sequences. To successfully use the machine learning methods to identify PPIs from primary protein amino acids sequences, one of the most important computational challenges is how to effectively represent a protein sequence by a fixed length feature vector in which the important information content of proteins is fully encoded [36, 37]. In this study, two kinds of sequence representation approach are used to transform the protein sequences into feature vectors, including amino acid composition and a novel local descriptor. For amino acid composition, it is evident that 20 amino acid composition descriptors reflecting the fraction of each kind of amino acid in a protein sequence are directly calculated. Then, a local multiscale decomposition technique is used to divide protein sequence into multiple sequence segments of varying length to describe local regions. Here, the continuous sequence segments are composed of residues which are local in the polypeptide sequence [38].

In order to extract local information, we first divided the entire protein sequence into seven equal length fractions. Then a novel binary coding scheme was adopted to construct a set of continuous regions on the basis of the above partition. For example, consider a protein sequence "CCYGGGYY-CYYYCGGCCYYCG" containing 21 residues. To represent the sequence by a feature vector, let us first divide each protein sequence into multiple regions. For simplicity, the protein sequence is divided into four equal length segments (denoted as S_1, S_2, S_3, and S_4). Then it is encoded as a sequence of 1's and 0's of 4-bit binary form. In binary format, these combinations are written as *0000*, *0001*, *0010*, *0011*, *0100*, *0101*, *0110*, *0111*, *1000*, *1001*, *1010*, *1011*, *1100*, *1101*, *1110*, and *1111*. The number of states of a group of bits can be found by the expression 2^n, where n is the number of bits. It should be noticed that here 0 or 1 denotes one of the four equal length regions, and S_1–S_4 are excluded or included in constructing the continuous regions, respectively. For example, 1100 denotes a continuous region constructed by S_1 and S_2 (the first 50% of the sequence). Similarly, 0011 represents a continuous region constructed by S_3 and S_4 (the final 50% of the sequence).

It should be noticed that the proposed representation can be simply and conveniently edited at multiple scales, which offers a promising new approach for addressing these difficulties in a simple, unified, and theoretically sound way when presenting a protein sequence. For a given number of bits, each protein sequence may take on only a finite number of continuous or discontinuous regions. This limits the resolution of the sequence. If more bits are used for each protein sequence, then a higher degree of resolution is obtained. In this study, the protein sequence is encoded by 7-bit binary form; each protein sequence may take on 126 (2^7-2) different regions. Higher bit encoding requires more storage for data and requires more computing resource to process. In this study, only the continuous regions are used and the discontinuous regions are discarded.

For each continuous region, three types of descriptors, composition (C), transition (T), and distribution (D), are used to represent its characteristics. C denotes the amino acids number of a particular property (e.g., hydrophobicity) divided by the total amino acids number in a local region. T is the percentage frequency with which amino acids for a particular property are followed by protein amino acids of another property. D characterizes the chain length within which the first 25 percent, 50 percent, 75 percent, and 100 percent of the protein amino acids of a particular property are located, respectively [39].

The three descriptors can be calculated in the following ways. Firstly, in order to reduce the complexity inherent in the representation of the 20 standard protein amino acids, we firstly clustered them into seven clusters based on the volumes and dipoles of the side chains. Amino acids within the same groups likely involve synonymous mutations because of their similar characteristics [29]. The amino acids belonging to each group are shown in Table 1.

Then, every amino acid in each protein sequence is replaced by the index depending on its grouping. For example, protein sequence "CCYGGGYYCYYYCGGCCYYCG" is replaced by 773111337333711773371 based on this classification of amino acids (see Figure 2). There are six "1," eight "3," and seven "7" in this protein sequence. The composition for these

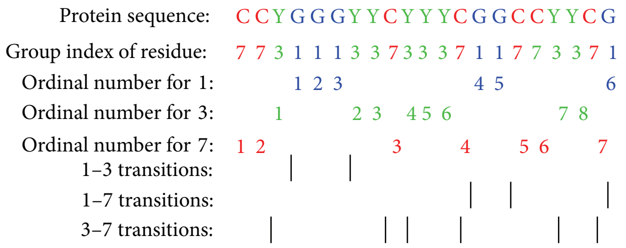
Protein sequence: C C Y G G G Y Y C Y Y Y C G G C C Y Y C G
Group index of residue: 7 7 3 1 1 1 3 3 7 3 3 3 7 1 1 7 7 3 3 7 1
Ordinal number for 1: 1 2 3 4 5 6
Ordinal number for 3: 1 2 3 4 5 6 7 8
Ordinal number for 7: 1 2 3 4 5 6 7
1–3 transitions:
1–7 transitions:
3–7 transitions:

Figure 2: Sequence of a hypothetic protein indicating the construction of composition, transition, and distribution descriptors of a protein region.

Table 1: Division of amino acids into seven groups based on the dipoles and volumes of the side chains.

Group	Class	Dipole scale	Volume scale
1	Ala, Gly, Val	Dipole < 1.0	Volume < 50
2	Ile, Leu, Phe, Pro	Dipole < 1.0	Volume > 50
3	Tyr, Met, Thr, Ser	1.0 < dipole < 2.0	Volume > 50
4	His, Asn, Gln, Trp	2.0 < dipole < 3.0	Volume > 50
5	Arg, Lys	Dipole > 3.0	Volume > 50
6	Asp, Glu	Dipole > 3.0 (opposite orientation)	Volume > 50
7	Cys	1.0 < dipole < 2.0 (form disulphide bonds)	Volume > 50

three symbols is $6/(6+7+8) \times 100\% = 28.57\%$, $8/(6+7+8) \times 100\% = 38.10\%$, and $7/(6+7+8) \times 100\% = 33.33\%$, respectively. There are 2 transitions from “1” to “3” or from “3” to “1” in this sequence, and the percentage frequency of these transitions is $(2/20)\times 100\% = 10\%$. The transitions from “1” to “7” or from “7” to “1” in this sequence can similarly be calculated as $(3/20) \times 100\% = 15\%$. The transitions from “3” to “7” or from “7” to “3” in this sequence can also similarly be calculated as $(6/20) \times 100\% = 30\%$.

For distribution D, there are 6 residues encoded as “1” in the example of Figure 3, the positions for the first residue “1,” the 2nd residue “1” ($25\% \times 6 = 2$), the 4th “1” residue ($50\% \times 6 = 3$), the 6th “1” ($75\% \times 6 = 5$), and the 8th residue “1” ($100\% \times 6 = 6$) in the encoded sequence are 4, 5, 6, 15, and 21, respectively, so the D descriptors for “1” are $(4/21) \times 100\% = 19.05\%$, $(5/21) \times 100\% = 23.81\%$, $(6/21) \times 100\% = 28.57\%$, $(15/21) \times 100\% = 71.43\%$, and $(21/21) \times 100\% = 100\%$, respectively. Similarly, the D descriptor for “3” and “7” is 14.29%, 33.33%, 47.62%, 57.14%, and 90.48% and 4.76%, 9.52%, 61.9%, 76.19%, and 95.24%, respectively.

For each continuous local region, the three descriptors (C, T, and D) were calculated and concatenated, and a total of 63 descriptors are generated: 7 for C, 21 ($(7 \times 6)/2$) for T, and 35 (7×5) for D. Then, the local descriptor from 27 regions (7-bit) was concatenated and a total 1701 dimensional vector has been built to represent each protein sequence. Finally, the PPI pair is characterized by concatenating the local and global descriptors of two individual proteins. Thus, a 3442-dimensional vector has been constructed to represent each protein pair and was used as a feature vector for input into SVM classifier.

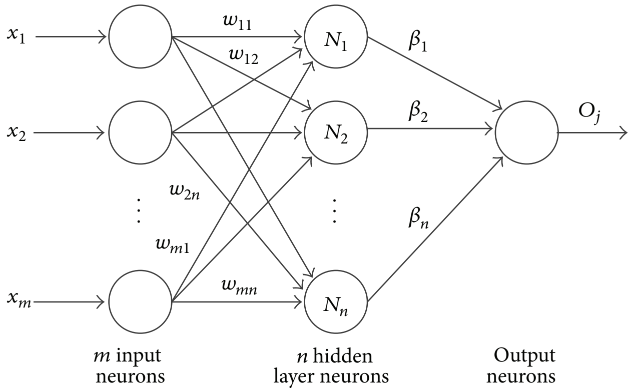

Figure 3: The structure of extreme learning machine.

2.3. Extreme Learning Machine.

By virtue of their approximation capabilities for nonlinear mappings, the feed-forward neural networks (FNN) have become ideal classifiers in many applications. Huang et al. proved that the single-hidden-layer FNN could exactly learn M distinct observations for almost any nonlinear activation function with almost M hidden nods [34, 40, 41]. However, the hidden layer biases and input weights of FNN have usually to be tuned using some parameter adjusting approach, which are generally time-consuming due to inappropriate learning steps with significantly large latency to converge to local maxima. Therefore, the slow learning speed of FNN has been a major bottleneck in different applications.

Extreme learning machine (ELM) was originally developed for the single hidden layer feed-forward neural network (SLFNN) and then extended to the generalized SLFNN where the hidden layer need not be neuron alike [34, 40]. As shown in Figure 3, its architecture is similar to that of a SLFNN. Recently the ELM algorithm has been increasingly popular in classification tasks due to its high generalization ability and fast learning speed. Different from the popular thinking that network parameters need to be adjusted, the input weights and first hidden layer biases need not be adjusted but they are randomly assigned in ELM. It has been proved that the ELM algorithm performs learning at an extremely fast speed and achieves a good generalization performance with activation functions which are infinitely differentiable in hidden layers [40, 42, 43].

The ELM algorithm transforms the learning problem into a simple linear system; that is, the output weights of ELM can be analytically determined through a generalized inverse operation of the hidden layer weight matrices. Compared with traditional learning frameworks such a learning scheme can operate at extremely much fast speed. Improved generalization performance of ELM with the smallest training error shows its superior classification capability for real-time applications at an exceptionally fast pace without any learning bottleneck [44].

The basic idea behind ELM algorithm is briefly descripted as follows: suppose learning N arbitrary distinct samples $(x_i, t_i) \in R^n \times R^m$, where $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T \subseteq R^n$, $t_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^T \subseteq R^m$, a standard ELM with L hidden neurons and activation function $g(x)$ are mathematically modeled by

$$\sum_{i=1}^{L} \beta_i g(x_j) = \sum_{i=1}^{L} \beta_i g(w_i \cdot x_j + b_i) = o_j, \quad j = 1, \ldots, N, \tag{1}$$

where $w_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^T$ represents the weight vector connecting the ith hidden node and the input nodes, $\beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^T$ represents the weight vector connecting the ith hidden neuron and the output neurons, and b_i is the bias of the ith hidden neuron. $w_i \cdot x_j$ denotes the inner product of w_i and x_j. A wide variety of functions could be selected as the activation function, including sigmoid function, radial basis function, sine function, hardlim function, and triangular basis function. The architecture of ELM is shown in Figure 3. Equation (1) can be written compactly as

$$H\beta = T, \tag{2}$$

where

$$H(w_1, \ldots, w_L, b_1, \ldots, b_L, \ldots, x_1, \ldots, x_N) = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_L \cdot x_1 + b_L) \\ \vdots & \cdots & \vdots \\ g(w_1 \cdot x_N + b_1) & \cdots & g(w_L \cdot x_N + b_L) \end{bmatrix}_{N \times L} \tag{3}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}, \qquad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m}.$$

H is termed as the hidden layer output matrix of the SLFNN; the ith column of H is the ith hidden neuron's output vector with respect to inputs $x_1, x_2, \ldots, x_N$. Hence for fixed arbitrary input weights w_i and the hidden layer bias b_i, training a SLFNN equals finding a least-squares solution $\widehat{\beta}$ of the linear system $H\beta = T$; that is,

$$\left\| H(w_1, \ldots, w_L, b_1, \ldots, b_L, x_1, \ldots, x_N) \widehat{\beta} - T \right\| = \min_{\beta} \left\| H(w_1, \ldots, w_L, b_1, \ldots, b_L, x_1, \ldots, x_N) \beta - T \right\|. \tag{4}$$

Equation (12) becomes a linear system and the solution is estimated as

$$\widehat{\beta} = H^{\dagger} T, \tag{5}$$

where $H^{\dagger}$ is the Moore-Penrose generalized inverse of the hidden layer output matrix H.

In summary, given a training dataset $\aleph = \{(x_i, t_i) \mid x_i \in R^n, t_i \in R^m, i = 1, \ldots, N\}$, activation function $g(x)$, and hidden neuron number L, the ELM-based learning procedure can be summarized as follows.

Step 1. Assign arbitrary input weight w_i and bias b_i, $i = 1, \ldots, L$.

Step 2. Calculate the hidden layer output matrix H.

Step 3. According to (13), calculate the output weight β.

3. Results and Discussion

In this section, we describe our simulation methodology and present the experimental results that evaluate the effectiveness of our schemes. The proposed sequence-based PPI predictor was implemented using MATLAB platform. For ELM algorithm, the implementation by Zhu and Huang available from http://www.ntu.edu.sg/home/egbhuang was used. Regarding SVM, LIBSVM implementation available from http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html was utilized, which was originally developed by Chang et al. [33]. Tree kinds of kernel functions were chosen and the optimized parameters were obtained with a grid search approach. All the simulations were carried out on a computer with 3.1 GHz 2-core CPU, 8 GB memory, and Windows operating system.

3.1. Cross Validation and Performance Evaluation. In the study, fivefold cross-validation technique has been employed to evaluate the performance of the proposed model. In fivefold cross-validation technique, the whole dataset is randomly divided into five subsets, where each subset consists of nearly equal number of interacting and noninteracting protein pairs. Four subsets are used for training and the remaining set for testing. This process is repeated five times so that each subset is used once for testing. The performance of method is average performance of method on five sets.

Seven metrics have been used in the study to measure the predictive ability of the proposed method. The parameters are as follows: (1) the overall prediction accuracy (ACC) is the percentage of correctly identified interacting and noninteracting protein pairs; (2) the sensitivity (SN) is the percentage of correctly identified interacting protein pairs; (3) the specificity (SP) is the percentage of correctly identified noninteracting protein pairs; (4) the positive predictive value (PPV) is the positive prediction value; (5) the negative predictive value (NPV) is the negative prediction value; (6) the F-score is a weighted average of the PPV and sensitivity, where an F-score reaches its best value at 1 and worst score at 0; (7) Matthew's correlation coefficient (MCC) is a more stringent measure of prediction accuracy accounts for both under- and overpredictions. These parameters are defined as follows:

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

$$\mathrm{SN}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \tag{7}$$

$$\mathrm{SP}=\frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}, \tag{8}$$

$$\mathrm{PPV}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \tag{9}$$

$$\mathrm{NPV}=\frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FN}}, \tag{10}$$

$$F1=2\times\frac{\mathrm{SN}\times\mathrm{PPV}}{\mathrm{SN}+\mathrm{PPV}}, \tag{11}$$

$$\begin{aligned}\mathrm{MCC}=&(\mathrm{TP}\times\mathrm{TN}-\mathrm{FP}\times\mathrm{FN})\\&\times((\mathrm{TP}+\mathrm{FN})\times(\mathrm{TN}+\mathrm{FP})\\&\times(\mathrm{TP}+\mathrm{FP})\times(\mathrm{TN}+\mathrm{FN}))^{-1/2},\end{aligned} \tag{12}$$

where true positive (TP) is the number of true PPIs that are predicted correctly; false negative (FN) is the number of true PPIs that are predicted to be noninteracting pairs; false positive (FP) is the number of true noninteracting pairs that are predicted to be PPIs, and true negative (TN) is the number of true noninteracting pairs that are predicted correctly.

The above mentioned parameters rely on the selected threshold. The area under the ROC curve (AUC), which is threshold-independent for evaluating the performances, can be easily calculated according to the following formula [45]:

$$\mathrm{AUC}=\frac{S_0-n_0(n_0+1)/2}{n_0\times n_1}, \tag{13}$$

where n_0 and n_1 denote the number of positive and negative samples, respectively, and S_0 is the sum of the ranks of all positive samples in the list of all samples ranked in increasing order by estimated probabilities belonging to positive. AUC values can give us a good insight into performance comparison of different prediction methods. Although the AUC is threshold-independent, an appropriate threshold must be selected for the final decision. For the classifier which outputs a continuous numeric value to represent the confidence or probability of a sample belonging to the predicted class, adjusting the classification threshold will lead to different confusion matrices which decide different ROC points [29].

3.2. Determination of ELM Parameter. The number of hidden nodes is a critical factor for the generalization of ELM. To determine the parameter, four-fifths of the whole dataset are randomly chosen to train the ELM classifiers with different number of hidden nodes, while the rest one-fifths of the dataset are used as the validation set to compute the accuracy.

Here the sigmoid function was used as the activation function of the ELM classifier. The results are plotted in Figure 4, which shows that the accuracy value reaches about 0.9 and increases slowly when the number of hidden neurons was set to 9 percent of the amount of samples. Based on Figure 4, we finally set 9 percent of the sample number as the number of hidden neurons for the ELM classifier. The second experiment was to examine how the running time scales with the number of hidden neurons. We increase the number of hidden neurons from 1 to 11 percent of the amount of samples and measure the average time overhead. Figure 5 shows that the running time of proposed ELM model scales nearly linear as the hidden neuron size increases.

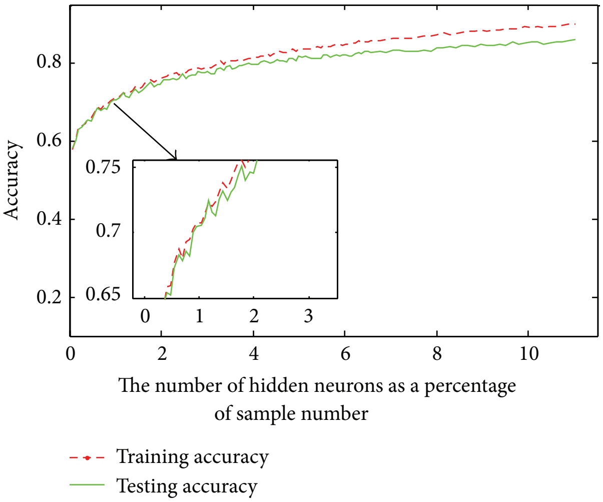

FIGURE 4: The relationship between the prediction accuracy and the number of hidden neurons. The x-axis denotes the number of hidden neurons as a percentage of sample number and the y-axis is the corresponding accuracy values.

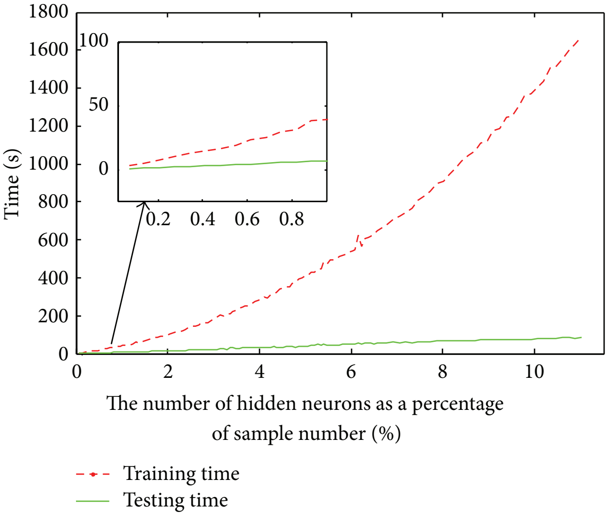

FIGURE 5: The relationship between the consuming time and the number of hidden neurons. The x-axis denotes the number of hidden neurons as a percentage of sample number and the y-axis is the running time.

3.3. Prediction Performance of Proposed Model. We evaluated the performance of the proposed model using the PPIs dataset as described in the aforementioned section. To guarantee that the experimental results are valid and can be generalized for making predictions regarding new data, we adopted the fivefold cross-validation in this study. The advantages of cross-validation are that the impact of data

Table 2: Comparison of the prediction performance by the proposed method and state-of-the-art SVM classifier on the human dataset.

Method	Kernel	Mean/std	Time (s)	ACC	SN	SP	PPV	NPV	F1	MCC	AUC
							Testing				
ELM	Sigmoid	Mean	72.7901	0.8480	0.8408	0.8553	0.8547	0.8415	0.8477	0.7422	0.9232
		Variance	1.9062	0.0022	0.0019	0.0028	0.0040	0.0038	0.0029	0.0030	0.0028
	Hardlim	Mean	77.4139	0.8206	0.8171	0.8242	0.8227	0.8185	0.8199	0.7056	0.9020
		Variance	3.7710	0.0050	0.0040	0.0063	0.0088	0.0026	0.0063	0.0064	0.0031
	Gaussian	Mean	76.9615	0.7257	0.7328	0.7186	0.7232	0.7283	0.7279	0.6018	0.7624
		Variance	4.1012	0.0036	0.0048	0.0054	0.0085	0.0077	0.0044	0.0033	0.0017
							Training				
ELM	Sigmoid	Mean	1282.12	0.8887	0.8831	0.8944	0.8933	0.8843	0.8882	0.8022	0.9561
		Variance	17.25	0.0006	0.0010	0.0018	0.0014	0.0001	0.0008	0.0010	0.0012
	Hardlim	Mean	1330.33	0.8668	0.8655	0.8682	0.8683	0.8654	0.8669	0.7691	0.9397
		Variance	46.28	0.0027	0.0021	0.0033	0.0027	0.0027	0.0024	0.0039	0.0031
	Gaussian	Mean	1435.45	0.7824	0.7896	0.7753	0.7790	0.7860	0.7843	0.6595	0.8626
		Variance	94.85	0.0033	0.0022	0.0053	0.0040	0.0026	0.0029	0.0037	0.0038
							Testing				
SVM	Sigmoid	Mean	2794.29	0.8177	0.8119	0.8232	0.8215	0.8144	0.8165	0.7018	0.8878
		Variance	16.71	0.0127	0.0266	0.0128	0.0067	0.0200	0.0155	0.0160	0.0143
	Gaussian	Mean	5237.89	0.6947	0.4714	0.9191	0.8535	0.6348	0.6064	0.5320	0.8997
		Variance	67.82	0.0228	0.0412	0.0112	0.0178	0.0265	0.0340	0.0276	0.0364
	Polynomial	Mean	3612.98	0.8019	0.8219	0.7819	0.7903	0.8144	0.8057	0.6820	0.8838
		Variance	20.16	0.0101	0.0126	0.0117	0.0165	0.0114	0.0125	0.0122	0.0138

dependency is minimized and the reliability of the results can be improved.

The prediction performance of ELM predictor with novel representation of protein sequence across five runs is shown in Table 2. It can be observed from Table 2 that high prediction accuracy of 84.8% is achieved for the ELM model with sigmoid function. To better investigate the prediction ability of our model, we also calculated the values of sensitivity, specificity, PPV, NPV, F-score, MCC, and AUC. From Table 2, we can see that our model gives good prediction performance with an average sensitivity value of 84.08%, specificity value of 85.53%, PPV value of 85.47%, NPV value of 84.15%, F-score value of 84.77%, MCC value of 74.22%, and AUC value of 0.9232. Further, it can also be seen in Table 2 that the standard deviation of accuracy, sensitivity, specificity, PPV, NPV, F-score, MCC, and AUC is as low as 0.0022, 0.0019, 0.0028, 0.0040, 0.0038, 0.0029, 0.0030, and 0.0028, respectively.

To demonstrate the performance of the proposed model, we further compared our method with the state-of-the-art predictor SVM. From Table 2, we can see the performance of ELM and SVM model. As observed from Table 2, the testing time of SVM algorithm (2794.29 s) is roughly 38 times the testing time of ELM algorithm (72.7901 s) for sigmoid activation function. In addition, the prediction performance of ELM is also promising. The AUC of the SVM algorithm is 0.8878, which is lower than the ELM. The overall accuracy, sensitivity, specificity, PPV, NPV, $F1$ score, and MCC of SVM algorithm are, respectively, 81.77%, 81.19%, 82.32%, 82.15%, 81.44%, 81.65%, and 70.18% as illustrated in Table 2. Hence, it can be seen that almost all evaluation measures of ELM algorithm are a little better than those of SVM algorithm, while its learning speed is much more faster than SVM.

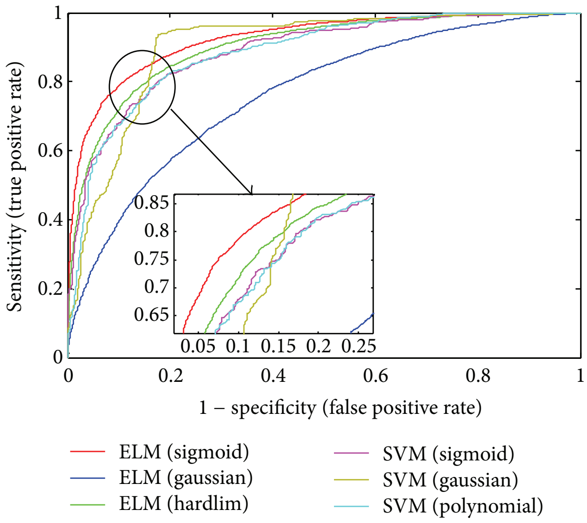

Figure 6: The ROC (receiver operator characteristic) curve illustrating the performance of different activation functions. The curve presents the true positive rate (sensitivity) against the false positive rate (1 − specificity).

We also conduct an experiment to characterize the sensitivity (i.e., the size of true positives that can be detected by our method) and specificity (i.e., 1 − false positive rate) of proposed approach for different activation functions

(Figure 6). The results in Figure 6 are reported using receiver operator characteristic (ROC) curves, which plot the achievable sensitivity at a given specificity (1 − false positive rate). Good performance is reflected in curves with a stronger bend towards the upper-left corner of the ROC graph (i.e., high sensitivity is achieved with a low false positive rate). We found that the proposed method achieved over 83 percent detection rate with less than 10 percent false positive rate. The results demonstrate that the proposed ELM can successfully classify positive and negative samples in all five activation functions that we investigated. Our algorithm can perfectly classify interacting and noninteracting protein pairs with only a few exceptions.

To sum up, considering the high efficiency as well as the good performance we can readily conclude that the proposed approach generally outperforms the state-of-the-art model with higher discrimination power for predicting PPIs based on the information of protein sequences. Therefore, we can see clearly that our model is a much more appropriate method for predicting new protein interactions compared with the other methods. Consequently, it makes us be more convinced that the proposed method can be very helpful in assisting the biologist to assist in the design and validation of experimental studies and for the prediction of interaction partners.

4. Conclusions

In this paper, we have developed an efficient and fast learning technique, which utilizes global and local information of protein amino acid sequence, for accurate identification PPIs at considerably high speed both in training and testing phase. The first contribution of this work is a novel protein amino acids sequence representation using amino acid composition and a descriptor to represent global and local information of a protein sequence, respectively. Then, the application of extreme learning machine ensures reliable recognition with minimum error and learning speed approximately thousands of times faster than the state-of-the-art classification method SVM. Experimental results demonstrated that the proposed method performed significantly well in distinguishing interacting and noninteracting protein pairs. It was observed that the proposed method achieved the mean classification accuracy of 84.8% using 5-fold cross-validation. Meanwhile, comparative study was conducted on the proposed method and the state-of-the-art SVM. The experimental results showed that our method significantly outperformed SVM in terms of classification accuracy with shorter running time.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Science Foundation of China, under Grants 61102119, 61373086, 61133010, U1201256, and 61171125. The authors would like to thank all the guest editors and anonymous reviewers for their constructive advices.

References

[1] M. Vestergaard, K. Kerman, and E. Tamiya, “An overview of label-free electrochemical protein sensors,” *Sensors*, vol. 7, no. 12, pp. 3442–3458, 2007.

[2] P. Uetz, L. Glot, G. Cagney et al., “A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae,” *Nature*, vol. 403, no. 6770, pp. 623–627, 2000.

[3] S. R. Collins, P. Kemmeren, X. Zhao et al., “Toward a comprehensive atlas of the physical interactome of *Saccharomyces cerevisiae*,” *Molecular and Cellular Proteomics*, vol. 6, no. 3, pp. 439–450, 2007.

[4] Y. Ho, A. Gruhler, A. Heilbut et al., “Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry,” *Nature*, vol. 415, no. 6868, pp. 180–183, 2002.

[5] N. Simonis, J. Rual, A. Carvunis et al., “Empirically controlled mapping of the caenorhabditis elegans protein-protein interactome network,” *Nature Methods*, vol. 6, no. 1, pp. 47–54, 2009.

[6] K. Venkatesan, J. Rual, A. Vazquez et al., “An empirical framework for binary interactome mapping,” *Nature Methods*, vol. 6, no. 1, pp. 83–90, 2009.

[7] H. Yu, P. Braun, M. A. Yildirim et al., “High-quality binary protein interaction map of the yeast interactome network,” *Science*, vol. 322, no. 5898, pp. 104–110, 2008.

[8] L. Giot, J. S. Bader, C. Brouwer et al., “A Protein Interaction Map of Drosophila melanogaster,” *Science*, vol. 302, no. 5651, pp. 1727–1736, 2003.

[9] V. Schachter, “Construction and prediction of protein: protein interaction maps,” in *Ernst Schering Research Foundation Workshop. Bioinformatics and Genome Analysis*, H. W. Mewes, H. Seidel, and B. Weiss, Eds., vol. 38, pp. 191–220, 2002.

[10] L. Giot, J. S. Bader, C. Brouwer et al., “A protein interaction map of Drosophila melanogaster,” *Science*, vol. 302, no. 5651, pp. 1727–1736, 2003.

[11] T. Huang, S. Wan, Z. Xu et al., “Analysis and prediction of translation rate based on sequence and functional features of the mRNA,” *PLoS ONE*, vol. 6, no. 1, Article ID e16036, 2011.

[12] Z. You, Y. Lei, J. Gui, D. Huang, and X. Zhou, “Using manifold embedding for assessing and predicting protein interactions from high-throughput experimental data,” *Bioinformatics*, vol. 26, no. 21, pp. 2744–2751, 2010.

[13] A. M. Edwards, B. Kus, R. Jansen, D. Greenbaum, J. Greenblatt, and M. Gerstein, “Bridging structural biology and genomics: assessing protein interaction data with known complexes,” *Drug Discovery Today*, vol. 9, no. 2, pp. S32–S40, 2004.

[14] G. Liu, J. Li, and L. Wong, “Assessing and predicting protein interactions using both local and global network topological metrics,” *Genome Informatics*, vol. 21, pp. 138–149, 2008.

[15] H. N. Chua and L. Wong, “Increasing the reliability of protein interactomes,” *Drug Discovery Today*, vol. 13, no. 15-16, pp. 652–658, 2008.

[16] T. Huang, J. Zhang, Z. Xu et al., “Deciphering the effects of gene deletion on yeast longevity using network and machine learning approaches,” *Biochimie*, vol. 94, no. 4, pp. 1017–1025, 2012.

[17] Z.-H. You, Y.-K. Lei, L. Zhu, J. Xia, and B. Wang, “Prediction of protein-protein interactions from amino acid sequences with

ensemble extreme learning machines and principal component analysis," *BMC bioinformatics*, vol. 14, supplement 8, article S10, 2013.

[18] T. Huang, C. Wang, G. Zhang, L. Xie, and Y. Li, "SySAP: a system-level predictor of deleterious single amino acid polymorphisms," *Protein and Cell*, vol. 3, no. 1, pp. 38–43, 2012.

[19] Z. You, Z. Yin, K. Han, D. Huang, and X. Zhou, "A semi-supervised learning approach to predict synthetic genetic interactions by combining functional and topological properties of functional gene network," *BMC Bioinformatics*, vol. 11, no. 1, article 343, 2010.

[20] T. Huang, J. Wang, Y. Cai, H. Yu, and K. Chou, "Hepatitis c virus network based classification of hepatocellular cirrhosis and carcinoma," *PLoS ONE*, vol. 7, no. 4, Article ID e34460, 2012.

[21] J. Song, Z. Yuan, H. Tan, T. Huber, and K. Burrage, "Predicting disulfide connectivity from protein sequence using multiple sequence feature vectors and secondary structure," *Bioinformatics*, vol. 23, no. 23, pp. 3147–3154, 2007.

[22] T. Huang, X. Shi, P. Wang et al., "Analysis and prediction of the metabolic stability of proteins based on their sequential features, subcellular locations and interaction networks," *PLoS ONE*, vol. 5, no. 6, Article ID e10972, 2010.

[23] J. Song, H. Tan, H. Shen et al., "Cascleave: towards more accurate prediction of caspase substrate cleavage sites," *Bioinformatics*, vol. 26, no. 6, Article ID btq043, pp. 752–760, 2010.

[24] T. Huang, K. Tu, Y. Shyr, C.-C. Wei, L. Xie, and Y.-X. Li, "The prediction of interferon treatment effects based on time series microarray gene expression profiles," *Journal of Translational Medicine*, vol. 6, article 44, 2008.

[25] L. Zhu, Z. You, D. Huang, and B. Wang, "t-LSE: a novel robust geometric approach for modeling protein-protein interaction networks," *PLoS ONE*, vol. 8, no. 4, Article ID e58368, 2013.

[26] Y. Guo, L. Yu, Z. Wen, and M. Li, "Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences," *Nucleic Acids Research*, vol. 36, no. 9, pp. 3025–3030, 2008.

[27] Y. Qi, J. Klein-Seetharaman, and Z. Bar-Joseph, "A mixture of feature experts approach for protein-protein interaction prediction," *BMC Bioinformatics*, vol. 8, no. 10, article S6, 2007.

[28] X.-Y. Pan, Y.-N. Zhang, and H.-B. Shen, "Large-scale prediction of human protein-protein interactions from amino acid sequence based on latent topic features," *Journal of Proteome Research*, vol. 9, no. 10, pp. 4992–5001, 2010.

[29] J. Shen, J. Zhang, X. Luo et al., "Predicting protein-protein interactions based only on sequences information," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 11, pp. 4337–4341, 2007.

[30] S. Pitre, M. Hooshyar, A. Schoenrock et al., "Short co-occurring polypeptide regions can predict global protein interaction maps," *Scientific Reports*, vol. 2, article 239, 2012.

[31] Y.-K. Lei, Z.-H. You, Z. Ji, L. Zhu, and D.-S. Huang, "Assessing and predicting protein interactions by combining manifold embedding with multiple information integration," *BMC Bioinformatics*, vol. 13, supplement 7, article S3, 2012.

[32] J. Song, H. Tan, M. Wang, G. I. Webb, and T. Akutsu, "Tangle: two-Level support vector regression approach for protein backbone torsion angle prediction from primary sequences," *PLoS ONE*, vol. 7, no. 2, Article ID e30361, 2012.

[33] C. Chang and C. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.

[34] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[35] P. Smialowski, P. Pagel, P. Wong et al., "The Negatome database: a reference set of non-interacting protein pairs," *Nucleic Acids Research*, vol. 38, no. 1, pp. D540–D544, 2009.

[36] J. Song, H. Tan, A. J. Perry et al., "PROSPER: an integrated feature-based tool for predicting protease substrate cleavage sites," *PLoS ONE*, vol. 7, no. 11, Article ID e50300, 2012.

[37] T. Huang, Z. Xu, L. Chen, Y. Cai, and X. Kong, "Computational analysis of HIV-1 resistance based on gene expression profiles and the virus-host interaction network," *PLoS ONE*, vol. 6, no. 3, Article ID e17291, 2011.

[38] T. Huang, M. Jiang, X. Kong, and Y. Cai, "Dysfunctions associated with methylation, microrna expression and gene expression in lung cancer," *PLoS ONE*, vol. 7, no. 8, Article ID e43441, 2012.

[39] I. Dubchak, I. Muchnik, S. R. Holbrook, and S. Kim, "Prediction of protein folding class using global description of amino acid sequence," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 92, no. 19, pp. 8700–8704, 1995.

[40] G. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 513–529, 2012.

[41] G. Huang, Q. Zhu, and C. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 1–4, pp. 985–990, July 2004.

[42] G. B. Huang, X. Ding, and H. Zhou, "Optimization method based extreme learning machine for classification," *Neurocomputing*, vol. 74, no. 1–3, pp. 155–163, 2010.

[43] G. Huang, M. Li, L. Chen, and C. Siew, "Incremental extreme learning machine with fully complex hidden nodes," *Neurocomputing*, vol. 71, no. 4–6, pp. 576–583, 2008.

[44] R. Minhas, A. A. Mohammed, and Q. M. Jonathan Wu, "A fast recognition framework based on extreme learning machine using hybrid object information," *Neurocomputing*, vol. 73, no. 10-12, pp. 1831–1839, 2010.

[45] D. J. Hand and R. J. Till, "A simple generalisation of the area under the ROC curve for multiple class classification problems," *Machine Learning*, vol. 45, no. 2, pp. 171–186, 2001.