Molecular BioSystems

PAPER

View Article Online
View Journal | View Issue

Check for updates

Cite this: *Mol. BioSyst.*, 2017, **13**, 1336

Predicting protein–protein interactions from protein sequences by a stacked sparse autoencoder deep neural network

Yan-Bin Wang,†[a] Zhu-Hong You, †*[a] Xiao Li,*[a] Tong-Hai Jiang,[a] Xing Chen, [b] Xi Zhou[a] and Lei Wang[a]

Received 29th March 2017,
Accepted 18th May 2017

DOI: 10.1039/c7mb00188f

rsc.li/molecular-biosystems

Protein–protein interactions (PPIs) play an important role in most of the biological processes. How to correctly and efficiently detect protein interaction is a problem that is worth studying. Although high-throughput technologies provide the possibility to detect large-scale PPIs, these cannot be used to detect whole PPIs, and unreliable data may be generated. To solve this problem, in this study, a novel computational method was proposed to effectively predict the PPIs using the information of a protein sequence. The present method adopts Zernike moments to extract the protein sequence feature from a position specific scoring matrix (PSSM). Then, these extracted features were reconstructed using the stacked autoencoder. Finally, a novel probabilistic classification vector machine (PCVM) classifier was employed to predict the protein–protein interactions. When performed on the PPIs datasets of *Yeast* and *H. pylori*, the proposed method could achieve average accuracies of 96.60% and 91.19%, respectively. The promising result shows that the proposed method has a better ability to detect PPIs than other detection methods. The proposed method was also applied to predict PPIs on other species, and promising results were obtained. To evaluate the ability of our method, we compared it with the-state-of-the-art support vector machine (SVM) classifier for the *Yeast* dataset. The results obtained *via* multiple experiments prove that our method is powerful, efficient, feasible, and make a great contribution to proteomics research.

1. Introduction

As the most important component of biological cells, proteins are essential for life activities because they perform most processes in the cell through a mutual effect. The interactive information between proteins is significantly important for the study of biological information. Recently developed high-throughput experimental methods, such as immunoprecipitation,[1] yeast two-hybrid screening methods,[2] and protein chip,[3] can be used to predict large scale PPIs. However, there are some unavoidable drawbacks of these experimental methods: they are time-consuming, have high-cost, and suffer from high rates of both false positive and false negative. Therefore, it is urgent to develop an effective machine learning method to improve the prediction accuracy of protein interactions. These methods can not only save time and cost, but also guide the function of large scale biological species. To date, owing to the emergence of machine learning technology, which is getting more and more developed, a large number of computational methods have been presented for the detection of PPIs based on different types of data including protein structure information, protein domains, phylogenetic profiles, and genomic information.[4–10] However, these methods cannot be widely used because the reliability and accuracy of these methods rely on the prior knowledge of protein pairs such as protein homology. To date, a great deal of PPI data obtained from different organisms has been generated, and many databases such as BIND,[11] DIP,[12] and MINT[13] have been established to store the PPI data. Compared to the rapid growth of protein sequences, other data types that can be used to predict the PPIs are limited. Therefore, the amino acid sequence-based method is widely used for the detection of proteins. For example, Ji-Yong An *et al.* have used the relevance vector machine combined with local phase quantization to predict the PPIs.[14] Bock and Gough developed a method using physiochemical descriptors and several structures *via* a support vector machine (SVM) to predict the PPIs.[15] Najafabadi *et al.* have proposed a method to solve this difficult problem using the Bayesian network.[16] Yan-zhi Guo used a support vector machine combined with autocovariance to predict the protein–protein interactions.[17] Shen *et al.* have adopted an SVM model to predict the PPI network by combining kernel

[a] *Xinjiang Technical Institutes of Physics and Chemistry, Chinese Academy of Science, Urumqi 830011, China. E-mail: zhuhongyou@ms.xjb.ac.cn, xiaoli@ms.xjb.ac.cn; Tel: +86-18160622862*

[b] *School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou, Jiangsu 221116, China*

† The first two authors should be regarded as joint first authors.

function of protein pairs with a conjoint triad feature.[18] Yu-An Huang *et al.* have used discrete cosine transformation with weighted sparse representation model to predict the PPIs.[19] In addition, many methods based on amino acid sequences have been reported in the literature.[20–25] Although these methods have made a breakthrough, there is still a scope for improving the efficiency and accuracy of the prediction model of PPIs.

In this study, we developed a novel computational approach to predict the PPIs from amino acid sequences using probabilistic classification vector machines model (PCVM) and combining Zernike moments descriptor with stacked autoencoder. This feature representation method combined with the PCVM classifier has remarkably performed in the detection of the PPIs. We adopted the Zernike moment feature representation on a position specific scoring matrix (PSSM) and then used a stacked autoencoder to reduce the influence of noise. Finally, PCVM, a powerful classifier, was used to obtain reliable results. More specifically, PSSM was adopted to represent each protein. Then, the Zernike moment descriptor was applied to capture the representative information of each protein PSSM and generate 1260-dimensional feature vector. Subsequently, as a method of deep-learning, autoencoder was adopted to reduce the dimensions of the feature and the influence of noise. Finally, we adopted the PCVM model to accomplish the classification. When the proposed method was applied on *Yeast* and *H. pylori* PPIs datasets and cross-species PPIs datasets, gratifying results were obtained. Compared to the state-of-the-art SVM, our proposed approach produces fantastic performance, which is better than that of SVM. These experimental results clearly show that the proposed approach is reliable and powerful for the prediction of PPIs.

2. Materials and methodology

2.1. Dataset

The proposed approach was verified on publicly available dataset: *Yeast* and *H. pylori*, which could be extracted from the Database of Interacting Proteins (DIP). The reliability of the experiment was ensured by extracting 5594 protein pairs that contained a protein with more than 50 residues or less than 40% sequence identity to constitute the positive dataset from the *Yeast* dataset. We extracted 5594 protein pairs whose subcellular localizations were different to constitute the negative protein dataset from the *Yeast* dataset. The whole dataset consisted of 11 188 protein pairs, where one half were the positive datasets and the other half were negative datasets. Analogously, we extracted 1458 positive protein pairs to constitute the positive dataset and 1458 negative protein pairs to constitute the negative protein dataset from the *H. pylori* dataset.

2.2. Position specific scoring matrix

Position specific scoring matrix (PSSM)[26,27] was first adopted to detect distantly related proteins. Currently, it is also used for the detection of protein quaternary structural attributes and protein folding patterns. Herein, we also employed PSSM to predict the PPIs. A PSSM for a protein sequence is a $N \times 20$ matrix $Q = \{Q_{ij}: i = 1\ldots N \text{ and } j = 1\ldots 20\}$ where N is the length of a protein sequence, 20 is the total number of amino acids. Q_{ij} of the PSSM element is an assigned score that represents j_{th} amino acid in the i_{th} position of the query protein sequence, where Q_{ij} is expressed as $Q_{ij} = \sum_{k=1}^{20} p(i,k) \times w(j,k)$, where $p(i,k)$ is the appearing frequency value of the k_{th} amino acid at the position i of the probe and $w(j,k)$ represents the value of Dayhoff mutation matrix between the j_{th} and the k_{th} amino acids.[28] Consequently, a high score means a well-conserved position and a low score means a weakly conserved position.

In our study, we used PSI-BLAST to transform each protein into a PSSM to build the experimental datasets for the detection of the PPIs. To obtain highly homologous sequences, we chose three iterations and set the *e*-value parameter of PSI-BLAST to 0.001.

2.3. Zernike moments

Zernike moments have a distinctive performance in the fields of image analysis and object recognition.[29–31] As a feature extraction method, it can represent information from multiple angles, regardless of the orientation of variations. In this study, Zernike moments were introduced to obtain significant knowledge from a protein sequence. We described Zernike moments and illustrated why it could reach the rotation invariance. Finally, the process of feature selection has also been described.

A. Introduction of Zernike moment descriptor. The coefficients of Zernike moments are the outputs of the expansion of a function $f(\rho,\theta)$ into a complete orthogonal set of complex basis functions $V_{nm}(x,y)$. Zernike moment magnitude is rotationally invariant and suitable for feature description. In two-dimensional space, $V_{nm}(x,y)$ with the order n and repetition m is defined over a unit circle in the polar coordinates as follows:

$$V_{nm}(x,y) = V_{nm}(\rho,\theta) = R_{nm}(\rho)\mathrm{e}^{jm\theta} \quad \text{for } \rho \leq 1 \tag{1}$$

where $n \geq 0$ and m is an integer subject to the constraints: $n - |m|$ is even and $|m| \leq n$. Herein, $\{R_{nm}(\rho)\}$is a radial polynomial in the form of

$$R_{nm}(\rho) = \sum_{s=0}^{(n-|m|/2)} (-1)^s \frac{(n-s)!}{s!\left(\dfrac{n+|m|}{2}-s\right)!\left(\dfrac{n+|m|}{2}-s\right)!}\rho^{n-2s} \tag{2}$$

Note that $R_{n,-m}(\rho) = R_{nm}(\rho)$. The set of polynomials $V_{nm}(x,y)$ is orthogonal, *i.e.*,

$$\int_0^{2\pi}\int_0^1 V_{nm}{}^*(\rho,\theta)V_{pq}(\rho,\theta)\rho\mathrm{d}\rho\mathrm{d}\theta = \frac{\pi}{n+1}\delta_{np}\delta_{mq} \tag{3}$$

with

$$\delta_{ab} = \begin{cases} 1 & a = b \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

For continuous function $f(\rho,\theta)$, the Zernike moments are denoted by

$$Z_{nm} = \frac{n+1}{\pi}\int_0^{2\pi}\int_0^1 f(\rho,\theta)V_{nm}{}^*(\rho,\theta)\rho\mathrm{d}\rho\mathrm{d}\theta \tag{5}$$

For a digital function, the Zernike moments are given as

$$Z_{nm} = \frac{n+1}{\pi} \sum_{(\rho,\theta)\in\text{unitcircle}} \sum f(\rho,\theta) V_{nm}{}^{*}(\rho,\theta) \tag{6}$$

To compute the Zernike moments, the center of the PSSM matrix is taken as the origin and coordinate mapped into a unit disk, *i.e.*, $x^2 + y^2 \leq 1$. Note that $Z_{nm}{}^{*} = Z_{n,-m}$.

B. Invariance of the Zernike moment. In the same polar coordinate, $f'(\rho,\theta)$ is defined as the rotated function, and the relationship between the rotated and original function is as follows:

$$f'(\rho,\theta) = f(\rho,\theta - \alpha) \tag{7}$$

The Zernike moment of the function in the same coordinate is

$$Z_{nm} = \frac{n+1}{\pi}\int_0^{2\pi}\int_0^1 f(\rho,\theta) V_{nm}{}^{*}(\rho,\theta)\rho \mathrm{d}\rho \mathrm{d}\theta \tag{8}$$

$$= \frac{n+1}{\pi}\int_0^{2\pi}\int_0^1 f(\rho,\theta) R_{nm}(\rho)\mathrm{e}^{-jm\theta}\rho \mathrm{d}\rho \mathrm{d}\theta \tag{9}$$

The Zernike moment of the rotated function in the same coordinate is

$$Z_{nm}{}' = \frac{n+1}{\pi}\int_0^{2\pi}\int_0^1 f(\rho,\theta-\alpha) R_{nm}(\rho)\mathrm{e}^{-jm\theta}\rho \mathrm{d}\rho \mathrm{d} \tag{10}$$

Via a change of variable $\theta_1 = \theta - \alpha$

$$Z_{nm}{}' = \frac{n+1}{\pi}\int_0^{2\pi}\int_0^1 f(\rho,\theta_1) R_{nm}(\rho)\mathrm{e}^{-jm(\theta+\alpha)}\rho \mathrm{d}\rho \mathrm{d}\theta \tag{11}$$

$$= \left[\frac{n+1}{\pi}\int_0^{2\pi}\int_0^1 f(\rho,\theta_1) R_{nm}(\rho)\mathrm{e}^{-jm\theta_1}\rho \mathrm{d}\rho \mathrm{d}\theta\right]\mathrm{e}^{-jm\alpha} \tag{12}$$

$$= Z_{nm}\mathrm{e}^{-\mathrm{jm}\alpha} \tag{13}$$

We can see that Z_{nm}' of the rotated function $f'(\rho,\theta)$ becomes

$$Z_{nm}' = Znme^{-jm\alpha} \tag{14}$$

From eqn (14), we found that Zernike moments only produce phase shift on rotation. Therefore, the magnitude of the Zernike moments $|Z_{nm}'|$ can be adopted since it is robust against the rotation feature.[32–36]

C. Feature selection. According to the abovementioned discussion, it is known that the magnitudes of the Zernike moments can be regarded as rotation-invariant features. One problem that must be considered is how big should be the value of N. The lower order moments extract gross information and detailed information, which were captured by higher order moments. In the present experiments, N was set to 70. We obtained 1260 features from each protein sequence. The feature vector $\vec{F}$ can be represented as follows:

$$\vec{F} = [|A_{11}|, |A_{22}|, \ldots, |A_{NM}|]^{\mathrm{T}} \tag{15}$$

where $|A_{nm}|$ represents the magnitude of the Zernike moments. Herein, we did not consider the case of $m = 0$ because it did not include useful information regarding the PPIs and Zernike moments without considering $m < 0$ since they were inferred through $A_{n,-m} = A_{nm}{}^{*}$.

2.4. Stacked autoencoder

Deep learning exhibits a record-breaking performance in different fields. A stacked autoencoder with multiple layers is a type of deep learning, used for feature reduction and reconstruction.

A stacked autoencoder is a neural network consisting of multiple layers of sparse autoencoders in which the outputs of each layer are wired to the inputs of the successive layers.[37–42] Furthermore, it often captures a kind of useful information and reconstructs input in output layers. To observe this, an autoencoder tends to learn the features that form a good representation of its input. The first layer of a stacked autoencoder is inclined to learn first-order features in the raw input, like edges in an image. The second layer of a stacked autoencoder is apt to learn the second-order features corresponding to the patterns in the appearance of first-order features. Higher layers of the stacked autoencoder tend to learn even higher-order features. We employed the stacked autoencoder to obtain more robust and stable feature from the Zernike moments. In this section, we described the principle of an autoencoder.

The stacked autoencoder is a neural network consisting of multiple layers of basic sparse auto encoder (SAE) in which the outputs of each layer are connected to the inputs of each successive layer.

The basic autoencoder model is a fully connected three-layer neural network, including the two stages of coding and decoding, as shown in Fig. 1. In Fig. 1, there is one hidden layer and the input x is mapped onto z in the coding stage. This part can be expressed as

$$z = \sigma_1(W_1x + b_1) \tag{16}$$

Herein, σ_1 is a nonlinear function. After this, z is mapped onto the reconstruction x' of the same shape as x:

$$x' = \sigma_2(W_2x + b_2) \tag{17}$$

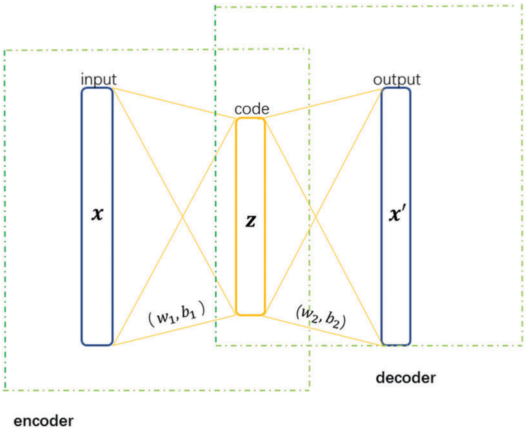

Fig. 1 Schematic of the architecture of a basic autoencoder.

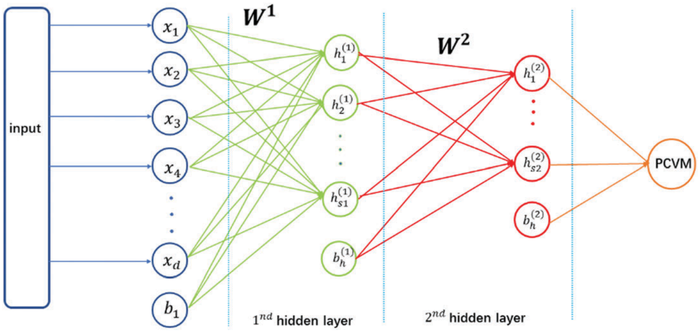

Fig. 2 Schematic of the architecture of a stacked sparse autoencoder.

where W_1 is the encoder weight matrix and b_1 is the bias and W_2 is the decoder weight matrix and b_2 is the bias. The objective of autoencoder is to minimize the reconstruction error between the input x and output y, with respect to the loss function L:

$$\theta = \arg\min \frac{1}{n}\sum_{i=1}^{n} L(x_i, y_i) \tag{18}$$

In this study, we considered the stacked sparse autoencoder (SSAE), which consists of two hidden layers. The architecture of SSAE is shown in Fig. 2.

For simplicity, we have not shown the decoder parts of SSAE in Fig. 2. Like SAE, training SSAE involves finding the best parameters while minimizing the difference between the input and its reconstruction. When the optimal parameters θ are obtained, the SSAE yields the function $R^{d_x} \to R^{d_{h^{(2)}}}$ that transforms input into a new feature representation $h^{(2)}$, and the feature reduction and feature reconstruction procedures are completed. As observed in Fig. 2, the input x was reconstructed by a high-level structured representation in the second hidden layer of the model.

2.5. PCVM algorithm

In the prediction phase, probabilistic classification vector machine (PCVM) was adopted to accurately predict the interactions among proteins. The PCVM has many advantages and effectively overcome some defects of SVM and RVM.

SVM has some disadvantages: (1) the number of support vectors linearly increase with the size of the training set; (2) SVM requires a cross-validation procedure to estimate the kernel parameter, which wastes computation time. For addressing these drawbacks of SVM, relevance vector machines (RVM) have been proposed. The RVM based on Bayesian technique has been acknowledged as an excellent model that supervise learning in pattern recognition. The RVM model gives a zero-mean Gaussian prior over every weight w_i and takes advantage of Bayesian automatic relevance determination (ARD) framework to produce a sparse solution. However, whether in positive or negative classes, RVM uses zero-mean Gaussian prior over weights, thereby leading to a drawback that position weights may be given to some training points belonging to the negative classes and *vice versa*. Hence, the decision of RVM can be influenced by some unreliable vectors. For the purpose of solving the problem of RVM, the PCVM gave different priorities over weights for training samples that belong to different class, *i.e.*, non-positive and right-truncated Gaussian was used for negative class; the non-negative and left-truncated Gaussian was used for the positive class. PCVM provides the following advantages: (1) PCVM is a probabilistic model that creates probabilistic outputs for each test sample; (2) PCVM adopts the expectation-maximization (EM) algorithm to optimize the kernel parameters. This optimization method is more efficient than the grid search by cross validation; (3) PCVM generates a sparse model because the number of weight vector is less. The sparseness reduces the computational time in the testing stage.[43–45]

PCVM is a supervised classification model. A set of input-target training pairs $\{x_i, y_i\}_{i=1}^{N}$ is given to learn a classification model $f(x;w)$, which is controlled by the parameters W. Considering the two-class classification problem, $y_i = \{-1,+1\}$. The model $f(x;w)$ is a linear combination of N basis functions $\phi_{i,\theta}(x)$ and has the following form:

$$f(x;w) = \sum_{i=1}^{N} w_i \phi_{i,\theta}(x) + b = \Phi_\theta(x)w + b \tag{19}$$

where the $\Phi_\theta(x) = \{\phi_{1,\theta(x)},\ldots,\phi_{N,\theta}(x)\}$ represents the basis function vector, wherein θ represents the parameter vector of the basis function, the vector $W = (w_1,\ldots,w_N)^{\mathrm{T}}$ represents the parameter of the model, and b represents the bias. The PCVM model achieves good classification results by adjusting the parameters b, θ, and W.

To obtain the binary outputs, the probit link function $\psi(x) = \int_{-\infty}^{x} N(t|0,1)\mathrm{d}t$ is used to be incorporated into the kernel method. Then, the PCVM model becomes

$$L(X;w,b) = \psi\left(\sum_{i=1}^{N} w_i \phi_{i,\theta}(x) + b\right) = \psi(\Phi_\theta(X)W + b) \tag{20}$$

A truncated Gaussian distribution as prior is given to each weight w_i, and a zero-mean Gaussian distribution as prior is given over the bias b as follows:

$$p(W|\boldsymbol{\alpha}) = \prod_{i=1}^{N} p(w_i|\alpha_i) = \prod_{i=1}^{N} N_t\left(w_i|0, \alpha_i^{-1}\right) \tag{21}$$

$$p(b|\beta) = N(b|0,\beta^{-1}) \tag{22}$$

where $N_t(w_i|0,\alpha_i^{-1})$ represents a truncated Gaussian function, α_i stands for the precision of the corresponding parameter and w_i, and β represents the precision of the normal distribution of b. When $y_i = +1$, the truncated prior is left-truncated Gaussian, which is non-positive, and when $y_i = -1$, the prior is a right-truncated Gaussian, which is also non-positive. This can be formalized in (23):

$$p(w_i|\alpha_i) = \begin{cases} 2N\left(w_i|0,\alpha_i^{-1}\right) & y_i w_i \geq 0 \\ 0 & \text{others} \end{cases} \tag{23}$$

The gamma distribution is applied as the hyper-prior of α and β. We used the EM algorithm to specify the parameters of the PCVM model—b, W, and θ. The EM algorithm is an iterative method of finding maximum likelihood or maximum *a posteriori* (MAP) estimate of parameters in the models, where the model depends on the unobserved latent variables. For more details about the PCVM theory, please refer to the literature.[46–48]

2.6. Initial parameter selection and training

The PCVM algorithm only needs to input one parameter θ, which can be automatically optimized by EM algorithm in the model training process. However, the EM algorithm is easily affected by the initial value and may fill into the local maxima. To address this problem, we adopted a method wherein the EM algorithm was run multiple times from different initial values over the same dataset. The same procedure was used to select the best initialization point of PCVM. A PCVM model with seven initialization values would be preferred over five training folds of each dataset. Then, a 5 × 7 matrix of the parameter θ, where the columns represent the initializations and the rows represent the folds, was obtained. For each row, we selected the results with highest test accuracy. Hence, the results reduced from 35 to 5 elements, and then, we selected the medium over these parameters. *Via* this method, the optimal original value θ was obtained, and the θ of the *Yeast* dataset and *H. pylori* dataset was set to 3.6 and 0.88, respectively.

3. Results and discussion

3.1. Evaluation measure

To evaluate the proposed method, the reliability of the model was verified using 5-fold cross-validation. The five training sets and five test sets were obtained by performing five separate operations, each of which produced a training set and a test set. Therefore, five predicting models were built over five data sets. We obtained five models and results with the model-evaluation being performed five times. The proposed method of model-evaluating follows these criteria: accuracy, sensitivity, precision, and Matthew's correlation coefficient (MCC). All the computational formulas are defined as follows:

$$\mathrm{Ac} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}} \tag{24}$$

$$\mathrm{Sn} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{TN}} \tag{25}$$

Table 1 Five-fold cross validation results obtained using the proposed method on the *Yeast* dataset

Testing set	Acc (%)	Sen (%)	Pre (%)	MCC (%)
1	96.42	93.87	96.86	93.09
2	96.56	93.72	99.54	93.35
3	96.92	93.93	99.91	94.01
4	96.38	93.08	99.41	92.99
5	96.70	94.36	99.07	93.61
Average	96.60 ± 0.22	93.73 ± 0.46	99.36 ± 0.41	93.41 ± 0.41

Table 2 Five-fold cross validation results obtained using the proposed method on the *H. pylori* dataset

Testing set	Acc (%)	Sen (%)	Pre (%)	MCC (%)
1	91.08	89.45	91.45	83.68
2	92.28	89.61	95.50	85.73
3	89.88	87.93	91.40	81.79
4	91.08	88.09	92.78	83.65
5	91.61	90.26	93.60	84.60
Average	91.19 ± 0.88	89.07 ± 1.01	92.95 ± 1.70	83.89 ± 1.45

$$\mathrm{Pe} = \frac{\mathrm{TP}}{\mathrm{FP} + \mathrm{TP}} \tag{26}$$

$$\mathrm{Mcc} = \frac{(\mathrm{TP} \times \mathrm{TN}) - (\mathrm{FP} \times \mathrm{FN})}{\sqrt{(\mathrm{TP} + \mathrm{FN}) \times (\mathrm{TN} + \mathrm{FP}) \times (\mathrm{TP} + \mathrm{FP}) \times (\mathrm{TN} + \mathrm{FN})}} \tag{27}$$

where TP, TN, FP, and FN represent the total number of true positive, true negative, false positive, and false negative, respectively. In addition, the receiver operating characteristic (ROC)[49] curve and the area under an ROC curve (AUC)[50] were applied to evaluate the performance of the proposed method. The prediction result of this method on the *Yeast* and *H. pylori* dataset are shown in Tables 1 and 2.

3.2. Assessment of the prediction

From Tables 1 and 2, it can be seen that the proposed method for predicting PPIs yields satisfactory results on the *Yeast* and *H. pylori* datasets. On applying the proposed approach on the *Yeast* dataset, the average accuracy, precision, sensitivity, and MCC of 96.60%, 93.73%, 99.36%, and 93.41%, respectively, could be achieved. Their standard deviations were 0.22%, 0.46%, 0.41%, and 0.41%. Using the proposed approach on the *H. pylori* dataset, we also obtained good results of average accuracy, precision, sensitivity, and MCC of 91.19%, 89.07%, 92.95%, and 83.89%, and their standard deviations were 0.88%, 1.01%, 1.70%, and 1.45%, respectively. The ROC curves for the two datasets are shown in Fig. 3 and 4. We computed the AUC values for further evaluating the ability of the proposed approach whose average values for the *Yeast* and *H. pylori* datasets were 97.67% and 93.97%, respectively.

These criterion values were high but the standard deviations of these criterion were low. This was sufficient to show that the proposed approach was robust, accurate, and practical in predicting the PPIs. The possible reasons for the remarkable predictive ability lie in the highly discriminative features and the choice of the powerful PCVM classifier. The used feature

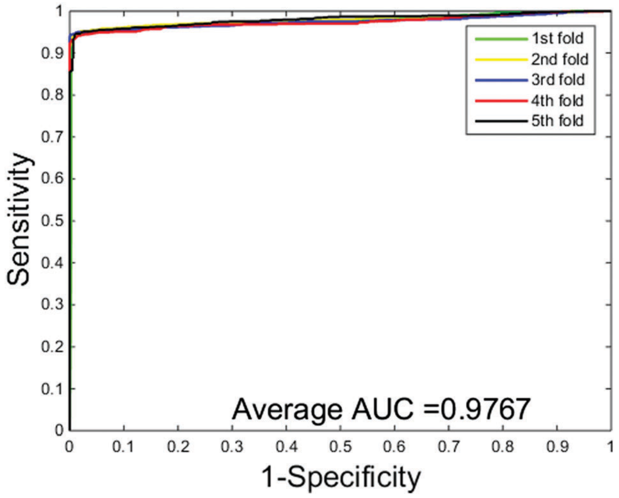

Fig. 3 ROC curves obtained for PCVM on the *Yeast* dataset.

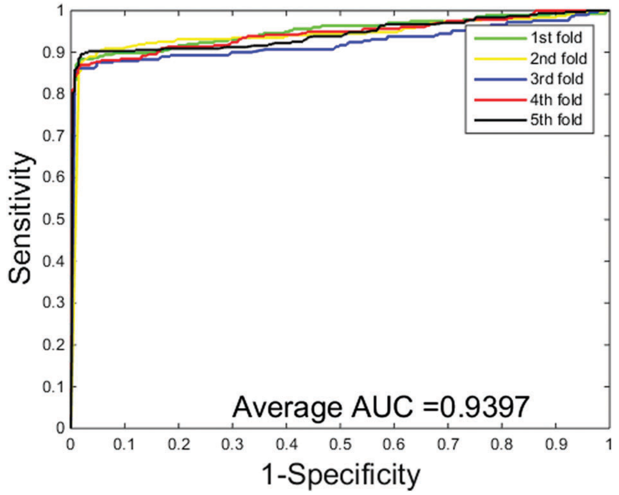

Fig. 4 ROC curves obtained for PCVM on the *H. pylori* dataset.

extraction method is effective and novel. The application of deep learning not only reduces the feature dimension, but also reconstructs the feature. As a representation of protein sequence, the PSSM preserves the probability of a given amino acid sequence at a specific location and has sufficient prior evolutionary information. The superior performance of the proposed approach can be attributed to the feature selection method and the classification model.

3.3. Comparison with the SVM-based method

To further verify the ability of the proposed method for the detection of the PPIs, the most widely used SVM model was adopted to predict the PPIs, and the results were compared with those of the PCVM model. To make it rational, same feature extraction methods were used in both models based on the *Yeast* dataset. The SVM could be used through the LIBSVM toolbox,[51] which was downloaded from https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/. We employed a grid search method to optimize the two parameters *c* and *g* of SVM, and the parameters *c* and *g* were set to 0.5 and 0.6, respectively. The radial basis function was chosen as the kernel function. For the PCVM and SVM classifiers, all the input vectors were normalized in the range of [−1, 1].

Table 3 Five-fold fold cross-validation results obtained using two model on the *Yeast* dataset

Model	Testing set	Acc (%)	Sen (%)	Pre (%)	MCC (%)
PCVM	1	96.42	93.87	98.86	93.09
	2	96.56	93.72	99.54	93.35
	3	96.92	93.93	99.91	94.01
	4	96.38	93.08	99.41	92.99
	5	96.70	94.36	99.07	93.61
	Average	96.60 ± 1.41	93.73 ± 1.05	99.36 ± 1.36	93.41 ± 1.16
SVM	1	94.86	90.26	99.31	90.20
	2	94.81	90.58	99.24	90.14
	3	94.95	90.63	99.22	90.37
	4	94.81	89.85	99.39	90.09
	5	95.58	92.07	99.15	91.54
	Average	95.00 ± 0.33	90.67 ± 0.84	99.26 ± 0.09	90.47 ± 0.61

The experimental results of these two methods are shown in Table 3, and the corresponding ROCs (receiver operating characteristic curve) are shown in Fig. 3 and 6. When PCVM classifier was used to detect the PPIs of the *Yeast* data set, great results were obtained with the average accuracy, sensitivity, precision, and MCC of 96.60%, 93.73%, 99.36%, and 93.41%, respectively. In contrast, SVM was found to be relatively poor with the average accuracy, sensitivity, precision, and MCC of 95.00%, 90.67%, 99.26%, and 90.47%, respectively, which demonstrated that the performance of PCVM was better than that of the SVM in the detection of the PPIs. Furthermore, the ROC curves of two methods on the H. Pylori data set are shown in Fig. 4 and 5. From Fig. 4 and 5, we can find that the ROC of the PCVM-based model is superior to that of the SVM-based classifier. We can observe that PCVM achieved higher average AUC than the SVM classifier. According to the case study, PCVM classifier was more accurate and efficient than SVM in detecting the PPIs. There are two main reasons for

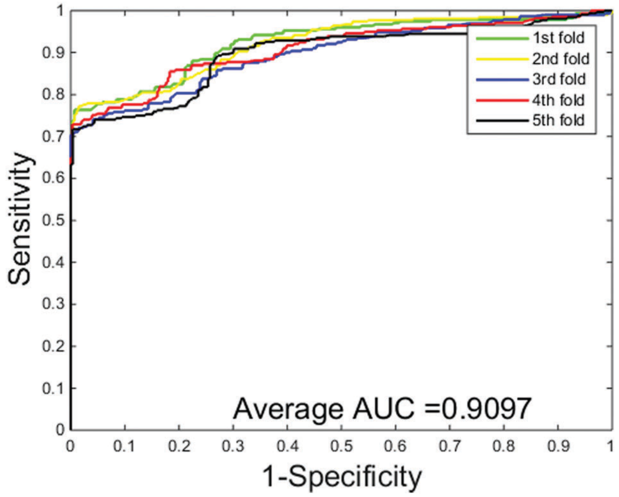

Fig. 5 ROC curves obtained for SVM on the *H. pylori* dataset.

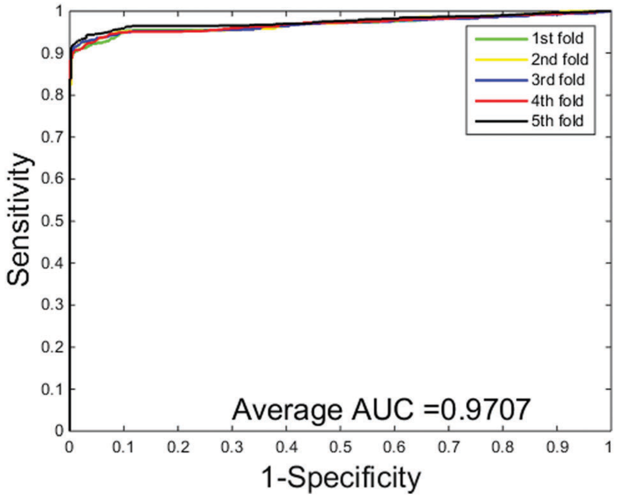

Fig. 6 ROC curves obtained for the SVM on the *Yeast* dataset.

improvement: (1) PCVM uses the truncated Gaussian priors to generate sufficient robustness and sparsity of the model, and these priors not only control the model complexity and reduce the time-computational amount in the test stage, but also improve the model generalization; (2) the PCVM employs an efficient kernel parameter optimization procedure, which is based on the probabilistic inference and EM algorithm. This process not only saves time for cross-validation grid search, but also improves the performance. It is for these reasons that the proposed method can achieve better results in the prediction of the PPIs.

3.4. Performance on independent data set

It can be obviously seen that the proposed method demonstrates strong ability for PPIs prediction on the *Yeast* and *H. pylori* datasets. However, we also conducted an extended analysis experiment to further verify the performance of this method on other species (Mix_Celeg, Mix_Ecoli, Mix_Hsapi, and Mix_Mmusc). In this experiment, a previous method was adopted to build a prediction model using the selected 11 188 samples of the Yeast datasets and the samples from other four species were used to test the prediction mode. The experiment results are shown in Table 4. On applying the proposed methods to predict the PPIs on four species, we achieved the high average prediction accuracy of 96.36%, 97.89%, 96.88%, and 97.12%. These promising results not only indicate that the Yeast proteins may have the interaction mechanisms similar to those of the other four species, but also suggest that the Yeast protein sequence can be used to predict the PPIs from other species. Moreover, it was observed that the proposed method had good generalization ability.

Table 4 Predictive results of the proposed method on four other species

Species	Test pairs	Accuracy (%)
Mix_Celeg	4013	96.36
Mix_Ecoli	6954	97.89
Mix_Hsapi	1412	96.88
Mix_Mmusc	313	97.12

4. Conclusion

Machine learning plays an important role in proteomics research, which can effectively improve the prediction accuracy of protein interaction. In this study, we explored a promising computational method to predict the PPIs using protein sequence only. The whole prediction model consists of the following stages. In the stage of feature selection, the Zernike moments were employed to extract the features from the PSSM matrix, which could effectively represent the evolutionary information of the protein sequence. In the stage of reducing the features, we adopted the deep learning method to reduce the feature dimensions. In the stage of classification, the PCVM model was used for predicting the PPIs. The experiments show that the proposed method has a strong ability to predict the PPIs.

Author contributions

Yan-Bin Wang, Zhu-Hong You, Xiao Li and Tong-Hai Jiang comprehended the algorithm, carried out the analyses, prepared the data sets, carried out the experiments and wrote the manuscript. Xing Chen, Xi Zhou and Lei Wang designed, performed, and analysed the experiments and wrote the manuscript. All the authors read and approved the final manuscript.

Conflicts of interest

The authors declare no conflicts of interest.

Acknowledgements

This work was partly supported by the National Science Foundation of China, under Grants 61373086, 11301517, 61572506, Guangdong Natural Science Foundation, under Grant 2014A030313555, and the Pioneer Hundred Talents Program of Chinese Academy of Sciences.

References

1 J. S. Bonifacino, E. C. Dell'Angelica and T. A. Springer, Immunoprecipitation. *Curr Protoc Neurosci*, 2006, ch. 7(1), Unit 7.2.

2 M. Koegl and P. Uetz, Improving yeast two-hybrid screening systems, *Briefings Funct. Genomics Proteomics*, 2007, **6**(4), 302–312.

3 H. Zhu and M. Snyder, Protein chip technology, *Curr. Opin. Chem. Biol.*, 2003, **7**(1), 55–63.

4 S. Qin and L. Cai, Predicting protein-protein interaction based on protein secondary structure information using Bayesian classifier, *J. Inn. Mong. Univ. Sci. Technol.*, 2010, **29**(1), 80–83.

5 G. Fernandezballester and L. Serrano, Prediction of protein-protein interaction based on structure, *Methods Mol. Biol.*, 2006, **340**, 207–234.

6 M. Boxem, Z. Maliga, N. Klitgord, N. Li, I. Lemmens, M. Mana, L. L. De, J. D. Mul, D. Van de Peut and M. Devos, A protein domain-based interactome network for *C. elegans* early embryogenesis, *Cell*, 2008, **134**(3), 534–545.

7 J. Xu, J. Liu, Z. Liu, Q. Shi and T. Li, Yixue: Refined phylogenetic profiles method for predicting protein-protein interactions, *Bioinformatics*, 2005, **21**(16), 3409.

8 T. Sato, Y. Yamanishi, M. Kanehisa, H. Toh and U. A. Jp, *Kyoto Tk: Prediction of protein–protein interactions based on real-valued phylogenetic profiles using partial correlation coefficient*, 2004.

9 A. Emamjomeh, B. Goliaei, A. Torkamani, R. Ebrahimpour, N. Mohammadi and A. Parsian, Protein-protein interaction prediction by combined analysis of genomic and conservation information, *Genes Genet. Syst.*, 2014, **89**(6), 259.

10 R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, S. Chung, A. Emili, M. Snyder, J. F. Greenblatt and M. Gerstein, A Bayesian networks approach for predicting protein-protein interactions from genomic data, *Science*, 2003, **302**(5644), 449.

11 G. D. Bader, I. Donaldson, C. Wolting, B. F. F. Ouellette, T. Pawson and C. W. V. Hogue, BIND—The Biomolecular Interaction Network Database, *Nucleic Acids Res.*, 2001, **29**(1), 242–245.

12 I. Xenarios, Ł. Salwínski, X. J. Duan, P. Higney, S. M. Kim and D. Eisenberg, DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions, *Nucleic Acids Res.*, 2002, **30**(1), 303–305.

13 L. Licata, L. Briganti, D. Peluso, L. Perfetto, M. Iannuccelli, E. Galeota, F. Sacco, A. Palma, A. P. Nardozza and E. Santonico, MINT, the molecular interaction database: 2012 update, *Nucleic Acids Res.*, 2007, **40**(suppl_1), D857–D861.

14 J. Y. An, F. R. Meng, Z. H. You, Y. H. Fang, Y. J. Zhao and Z. Ming, Using the Relevance Vector Machine Model Combined with Local Phase Quantization to Predict Protein-Protein Interactions from Protein Sequences, *BioMed Res. Int.*, 2016, **2016**(6868), 1–9.

15 J. R. Bock and D. A. Gough, Whole-proteome interaction mining, *Bioinformatics*, 2003, **19**(1), 125–134.

16 H. S. Najafabadi, Sequence-based prediction of protein-protein interactions by means of codon usage, *Genome Biol.*, 2008, **9**(5), 1–9.

17 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 Res.*, 2008, **36**(9), 3025.

18 J. Shen, J. Zhang, X. Luo, W. Zhu, K. Yu, K. Chen, Y. Li and H. Jiang, Predicting protein-protein interactions based only on sequences information, *Proc. Natl. Acad. Sci. U. S. A.*, 2007, **104**(11), 4337–4341.

19 Y. A. Huang, Z. H. You, G. Xin, W. Leon and L. Wang, 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.

20 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, *J. Proteome Res.*, 2010, **9**(10), 4992.

21 Z. H. You, L. Li, Z. Ji and M. Li, Prediction of protein-protein interactions from amino acid sequences using extreme learning machine combined with auto covariance descriptor, *Memetic Computing*, 2013, pp. 80–85.

22 Z. H. You, Y. K. Lei and L. Zhu, *et al.*, Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis[J], *BMC Bioinf.*, 2013, **14**(8), 1–11.

23 Z. H. You, K. C. Chan and P. Hu, Predicting protein-protein interactions from primary protein sequences using a novel multi-scale local feature representation scheme and the random forest.[J], *PLoS One*, 2015, **10**(5), e0125811.

24 Z. H. You, J. Li and X. Gao, *et al.*, Detecting Protein-Protein Interactions with a Novel Matrix-Based Protein Sequence Representation and Support Vector Machines[J], *BioMed Res. Int.*, 2015, **2015**(2), 1–9.

25 Z. H. You, Z. Ming and H. Huang, *et al.*, A novel method to predict protein–protein interactions based on the information of protein sequence[C]//IEEE International Conference on Control System, *IEEE Comput. Sci. Eng.*, 2012, 210–215.

26 J. C. Jeong, X. Lin and X. W. Chen, On Position-Specific Scoring Matrix for Protein Function Prediction, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 2011, **8**(2), 308–315.

27 N. Xiao, *Compute PSSM (Position-Specific Scoring Matrix) for given protein sequence*.

28 K. Tomii and M. Kanehisa, Analysis of amino acid indices and mutation matrices for sequence comparison and structure prediction of proteins, *Protein Eng.*, 1996, **9**(1), 27.

29 Z. Chen and S. K. Sun, A Zernike Moment Phase-Based Descriptor for Local Image Representation and Matching, *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, 2010, **19**(1), 205–219.

30 H. Hse and A. R. Newton, Sketched symbol recognition using Zernike moments, *International Conference on Pattern Recognition*, 2004, **1**(1), 367–370.

31 H. S. Kim and H. K. Lee, Invariant image watermark using Zernike moments, *IEEE Transactions on Circuits & Systems for Video Technology*, 2003, **13**(8), 766–775.

32 S. X. Liao and M. Pawlak, On the accuracy of Zernike moments for image analysis, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 1998, **20**(12), 1358–1364.

33 S. X. Liao and M. Pawlak, *A study of Zernike moment computing*, 2006.

34 R. Mukundan and K. R. Ramakrishnan, Fast computation of Legendre and Zernike moments, *Pattern Recogn.*, 1995, **28**(9), 1433–1442.

35 C. Singh, E. Walia and R. Upneja, Accurate calculation of Zernike moments, *Inf. Sci.*, 2013, **233**(233), 255–275.

36 J. L. Turney, T. N. Mudge and R. A. Volz, Invariant Image Recognition by Zernike Moments, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 1990, **12**(5), 489–497.

37 H. Liu, T. Taniguchi, T. Takano and Y. Tanaka, Visualization of driving behavior using deep sparse autoencoder, *Intelligent Vehicles Symposium Proceedings*, 2014, pp. 1427–1434.

38 Q. Xu and L. Zhang, The effect of different hidden unit number of sparse autoencoder, *Control and Decision Conference*, 2015, pp. 2464–2467.

39 Y. Bengio, Learning deep architectures for AI, *Journal Foundations and Trends in Machine Learning*, 2009, **2**(1), 1–127.

40 X. Pan, Y. X. Fan, J. Yan and H. B. Shen, IPMiner: hidden ncRNA-protein interaction sequential pattern mining with stacked autoencoder for accurate computational prediction, *BMC Genomics*, 2016, **17**(1), 582.

41 J. Xu, L. Xiang, R. Hang and J. Wu, Stacked Sparse Autoencoder (SSAE) based framework for nuclei patch classification on breast cancer histopathology, *IEEE International Symposium on Biomedical Imaging*, 2014, pp. 999–1002.

42 J. Xu, L. Xiang, Q. Liu, H. Gilmore, J. Wu, J. Tang and A. Madabhushi, Stacked Sparse Autoencoder (SSAE) for Nuclei Detection on Breast Cancer Histopathology images, *IEEE Transactions on Medical Imaging*, 2016, **35**(1), 119–130.

43 H. Chen, P. Tino and Y. Xin, Efficient Probabilistic Classification Vector Machine With Incremental Basis Function Selection, *IEEE Transactions on Neural Networks & Learning Systems*, 2014, **25**(2), 356–369.

44 R. Mohammadi, A. Mahloojifar, H. Chen and D. Coyle, *EEG Based Foot Movement Onset Detection with the Probabilistic Classification Vector Machine*, Springer, Berlin, Heidelberg, 2012.

45 F. M. Schleif, H. Chen and P. Tino, Incremental probabilistic classification vector machine with linear costs, *International Joint Conference on Neural Networks*, 2015.

46 H. Chen, P. Tino and X. Yao, Probabilistic classification vector machines, *IEEE Transactions on Neural Networks*, 2009, **20**(20), 901–914.

47 Z. Xue, X. Yu and Q. Fu *et al.*, Hyperspectral imagery classification based on probabilistic classification vector machines[C]//Eighth International Conference on Digital Image Processing, 2016, 100332C.

48 I. Jouny, Radar target identification using probabilistic classification vector machines[C]//SPIE Defense + Security. *International Society for Optics and Photonics*, 2016.

49 J. A. Hanley and B. J. Mcneil, The meaning and use of the area under a receiver operating characteristic (ROC) curve, *Radiology*, 1982, **143**(1), 29–36.

50 J. Huang and C. X. Ling, Using AUC and accuracy in evaluating learning algorithms, *Knowledge & Data Engineering IEEE Transactions on*, 2005, **17**(3), 299–310.

51 C. C. Chang and C. J. Lin, LIBSVM: A library for support vector machines, *Acm Transactions on Intelligent Systems & Technology*, 2007, **2**(3, article 27), 389–396.