



OPEN

## Robust and accurate prediction of protein–protein interactions by exploiting evolutionary information

Yang Li<sup>1</sup>, Zheng Wang<sup>1✉</sup>, Li-Ping Li<sup>1✉</sup>, Zhu-Hong You<sup>1</sup>, Wen-Zhun Huang<sup>1</sup>, Xin-Ke Zhan<sup>1</sup> & Yan-Bin Wang<sup>2</sup>

Various biochemical functions of organisms are performed by protein–protein interactions (PPIs). Therefore, recognition of protein–protein interactions is very important for understanding most life activities, such as DNA replication and transcription, protein synthesis and secretion, signal transduction and metabolism. Although high-throughput technology makes it possible to generate large-scale PPIs data, it requires expensive cost of both time and labor, and leave a risk of high false positive rate. In order to formulate a more ingenious solution, biology community is looking for computational methods to quickly and efficiently discover massive protein interaction data. In this paper, we propose a computational method for predicting PPIs based on a fresh idea of combining orthogonal locality preserving projections (OLPP) and rotation forest (RoF) models, using protein sequence information. Specifically, the protein sequence is first converted into position-specific scoring matrices (PSSMs) containing protein evolutionary information by using the Position-Specific Iterated Basic Local Alignment Search Tool (PSI-BLAST). Then we characterize a protein as a fixed length feature vector by applying OLPP to PSSMs. Finally, we train an RoF classifier for the purpose of identifying non-interacting and interacting protein pairs. The proposed method yielded a significantly better results than existing methods, with 90.07% and 96.09% prediction accuracy on *Yeast* and *Human* datasets. Our experiment show the proposed method can serve as a useful tool to accelerate the process of solving key problems in proteomics.

Proteins are the main functional components of biological cells, and they usually interact with DNA or other proteins in a specific way to perform their functions. Protein–protein interactions (PPIs) are critical to understanding the function of proteins and further manipulating many biological processes<sup>1</sup>. Therefore, the analysis of protein interactions has gradually become a hot topic in proteomics research. Thus far, researchers have discovered various experimental methods for detecting large-scale PPIs, including yeast two-hybrid<sup>2,3</sup>, protein chips<sup>4</sup>, tandem affinity purification<sup>5</sup>, immunoprecipitation<sup>6</sup>, and other high-throughput biotechnology. The rapid development of these high-throughput technologies has also accumulated available experimental data for the study of protein–protein interactions<sup>7</sup>. Nevertheless, biological experimental methods are expensive, time consuming, and labor intensive. Moreover, these methods typically perform poorly and are prone to produce low rates of true negative and true positive predictions<sup>8–10</sup>. Thus, an effective computational method to predict PPIs is highly desirable, and it may also alleviate the bottleneck of experimental methods<sup>11,12</sup>.

Currently, many computational methods based on various data types have been developed for predicting protein–protein interactions. The data sources involved in these methods mainly include literature mining knowledge<sup>13</sup>, gene fusion<sup>14</sup>, phylogenetic profiles<sup>15</sup>, gene ontology annotations<sup>16</sup>, gene neighborhood<sup>17</sup>, and co-evolution analysis of interacting proteins<sup>18</sup>. However, these methods are not commonly used to predict PPIs as they are difficult to apply if a priori information about the protein is not available. Moreover, the rapid development of genomic technology has led to an excessive accumulation of protein sequence data. Hence, it is very popular to predict protein–protein interactions based on protein sequence information<sup>19,20</sup>.

<sup>1</sup>School of Information Engineering, Xijing University, Xi'an 710123, China. <sup>2</sup>School of Cyber Science and Technology, Zhejiang University, Hangzhou 310027, China. ✉email: xywangzheng0971@163.com; Lipingli@gmail.com

Numerous previous studies have found that PPIs can be detected using only protein amino acid sequence data<sup>21,22</sup>. Guo et al.<sup>23</sup> reported a sequence-based method that combines auto-covariance (AC) and support vector machine (SVM) to predict PPIs. Among them, AC considers the neighbouring effect and explains the interaction between a certain number of residues in the sequence. The accuracy of this method on the *Saccharomyces cerevisiae* data was 88.09%. Pitre et al.<sup>24</sup> developed a computational engine called PIPE to predict protein–protein interactions. The engine can efficiently detect interactions among yeast protein pairs. The experimental results show that the PIPE algorithm achieves a sensitivity of 61% with 89% specificity and an average accuracy of 75% on yeast dataset. You et al.<sup>25</sup> proposed a hierarchical PCA-EELM method to predict PPIs, which utilizes only protein sequence information. Lei et al.<sup>26</sup> showed a neighbor affinity-based core-attachment method (NAB-CAM) to predict protein complexes from dynamic PPI networks. Huang et al.<sup>19</sup> presented a sequence-based substitution matrix representation (SMR) method to predict PPIs by using discrete cosine transform (DCT). This method yielded an average accuracy of 96.28% on the yeast dataset. Ding et al.<sup>27</sup> proposed a matrix-based protein sequence representation method that combines HOG and SVD feature representations as well as random forest classifiers to predict PPIs. Wang et al.<sup>28</sup> presented a computational model to predict PPIs, which is based on a Zernike moment (ZM) feature descriptor and a probabilistic classification vector machine (PCVM) algorithm. Although the existing prediction methods for protein–protein interactions have been developed by many investigators, there is still room for improvement in algorithms and prediction accuracy of PPIs.

In this paper, we report a protein sequence-based approach to detect protein–protein interactions. Specifically, all protein sequences were first converted to a position-specific scoring matrix (PSSM). Then, we use the orthogonal locality preserving projections (OLPP) algorithm to extract feature mathematical descriptors from each protein PSSM to obtain more representative information. Finally, we use the ensemble learning method in machine learning to perform the classification tasks of PPIs. The proposed method was applied to highly trusted *Yeast* and *Human* datasets to test the performance of PPIs prediction models. In addition, we demonstrate the predictive power of the proposed model on four separate datasets including *C. elegans*, *H. pylori*, *H. sapiens*, and *M. musculus*. Through further comparative experiments, our method obtains good prediction accuracy, which can reflect the reliability of the proposed method in predicting PPIs.

## Results and discussion

**Evaluation measures.** To validate the proposed model, we consider the following evaluation criteria in this experiment. The calculation formulas for overall prediction accuracy (Acc), precision (Pre), sensitivity (Sen), and Matthews correlation coefficient (MCC) are defined as:

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP}, \quad (1)$$

$$Precision = \frac{TP}{FP + TP}, \quad (2)$$

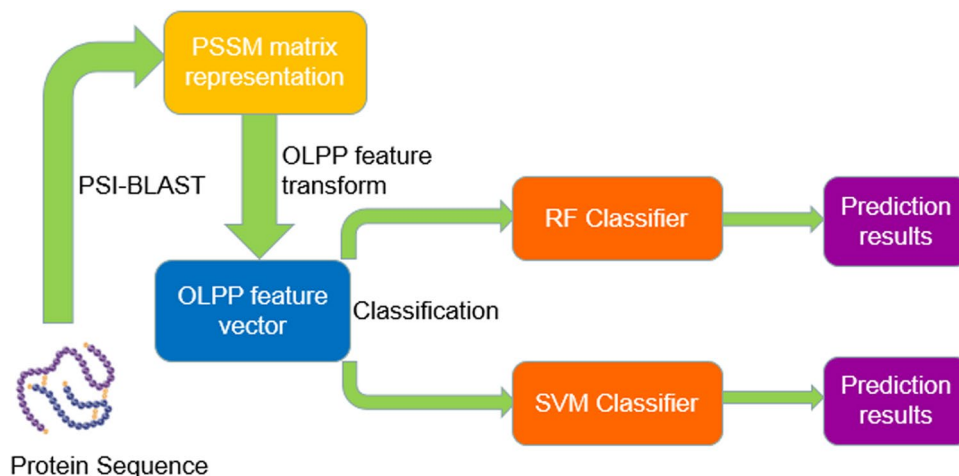
$$Sensitivity = \frac{TP}{TP + FN}, \quad (3)$$

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \times (TN + FN) \times (TN + FP) \times (TP + FN)}}, \quad (4)$$

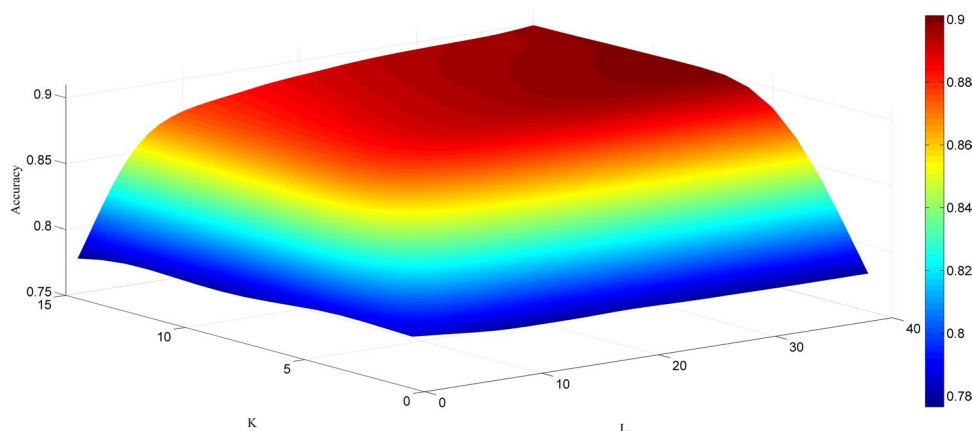
where *TN* is the number of true negatives, indicating that the non-interacting proteins are predicted correctly; *TP* is the amount of true positives, representing that the interacting proteins are predicted correctly; *FN* is the number of false negatives, indicating that the interacting proteins are predicted to be non-interacting; and *FP* is the amount of false positives, representing that the non-interacting proteins are predicted to have interaction. Additionally, the receiver operating characteristic (ROC)<sup>29</sup> curves and the area under the ROC curve (AUC)<sup>30</sup> were also calculated to further evaluate the discriminatory accuracy of the proposed model. The workflow of the proposed method is shown in Fig. 1.

**Assessment of prediction.** We applied the proposed method to two popular PPIs datasets to verify the performance of the model, including *Yeast* and *Human* datasets. In addition, to avoid over-fitting problems in the experiment, we used a five-fold cross-validation method to evaluate prediction performance. Specifically, we divided the entire dataset into five parts, four of which were used for training and one part was used for testing. In this way, we can obtain five separate models from the *Yeast* and *Human* datasets and perform five independent experiments. To be fair, we set the same parameters for the rotation forest classifier on different datasets. In this experiment, we use a grid search method to optimize two important parameters of the RoF algorithm. Figure 2 presents the prediction results of the RoF algorithm under different parameters. Here, the parameter *K* (the amount of feature subsets) is set to 10 and the parameter *L* (the amount of decision trees) is set to 35. The predicted results obtained by combining the proposed model with the five-fold cross-validation method on different datasets are shown in Table 1.

From Table 1, we can see that the proposed method for predicting PPIs has a good performance on the *Yeast* dataset. Its average accuracy, precision, sensitivity, and MCC were 90.07%, 90.24%, 89.83%, and 82.10%, respectively, and their standard deviations were 0.60%, 0.56%, 1.41%, and 0.97%, respectively. In addition, our method also achieved satisfactory results on the *Human* dataset. Its average accuracy, precision, sensitivity, and



**Figure 1.** The workflow of the proposed method.



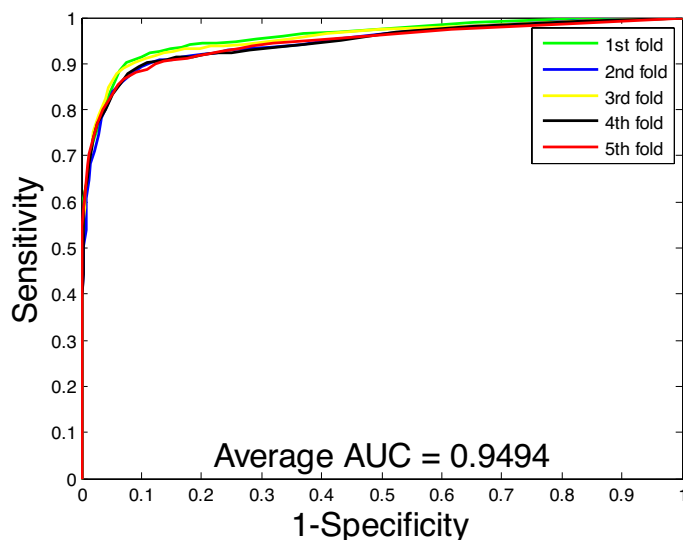
**Figure 2.** The accuracy surface obtained from the RoF algorithm for optimizing parameters  $K$  and  $L$ .

Data sets	Acc (%)	Pre (%)	Sen (%)	MCC (%)
<i>Yeast</i>	90.07 ± 0.60	90.24 ± 0.56	89.83 ± 1.41	82.10 ± 0.97
<i>Human</i>	96.09 ± 0.24	96.56 ± 0.36	95.20 ± 0.34	92.47 ± 0.46

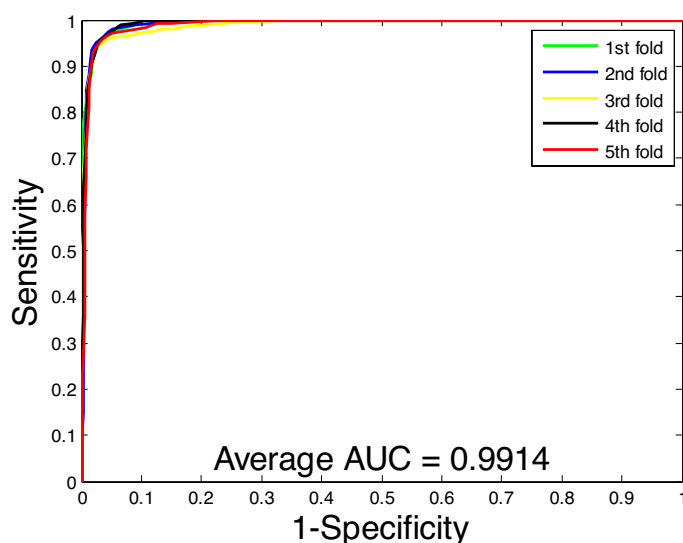
**Table 1.** Five-fold cross-validation prediction results using the proposed method on two dataset.

MCC were 96.09%, 96.56%, 95.20%, and 92.47%, respectively, and the standard deviations were 0.24%, 0.36%, 0.34%, and 0.46%, respectively. Figures 3 and 4 show the ROC curves of the proposed method on these two datasets, respectively. In the figure, the Y-axis refers to the true positive rate (TPR) and the X-axis refers to the false positive rate (FPR). To further evaluate the performance of the RoF classifier, we also obtained average AUC values of 94.94% and 99.14% on the *Yeast* and *Human* datasets, respectively. Observing these results, our method can achieve higher accuracy and lower standard deviation. This further indicates that the proposed method can effectively detect PPIs.

**Comparison of proposed method and support vector machine method.** Many algorithms and knowledge about machine learning are used to detect PPIs. Among them, support vector machine (SVM) is a popular supervised learning algorithm. To evaluate the predictive ability of the proposed model, we used the same feature extraction method to compare the prediction results of the two classifiers including RoF and SVM on *Yeast* and *Human* datasets. In this experiment, we use the LIBSVM tools as an SVM classifier, which can be downloaded from <https://www.csie.ntu.edu.tw/~cjlin/libsvm/>. To improve the prediction results of the SVM classifier on these two datasets, we use a grid search method to select two important parameters of SVM, namely the regularization parameter  $c$  and the kernel parameter  $g$ . When predicting PPIs on the *Yeast* dataset,



**Figure 3.** ROC curves performed using the proposed method on *Yeast* dataset.



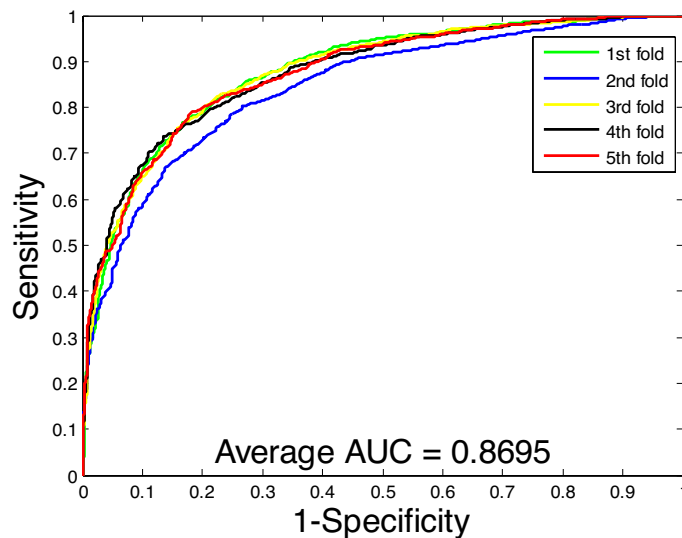
**Figure 4.** ROC curves performed using the proposed method on *Human* dataset.

Data sets	Classifier	Acc (%)	Pre (%)	Sen (%)	MCC (%)
<i>Yeast</i>	SVM	78.96 ± 1.55	79.08 ± 1.03	78.76 ± 2.37	66.80 ± 1.75
	RoF	90.07 ± 0.60	90.24 ± 0.56	89.83 ± 1.41	82.10 ± 0.97
<i>Human</i>	SVM	87.23 ± 0.57	87.23 ± 0.58	85.83 ± 1.16	77.66 ± 0.87
	RoF	96.09 ± 0.24	96.56 ± 0.36	95.20 ± 0.34	92.47 ± 0.46

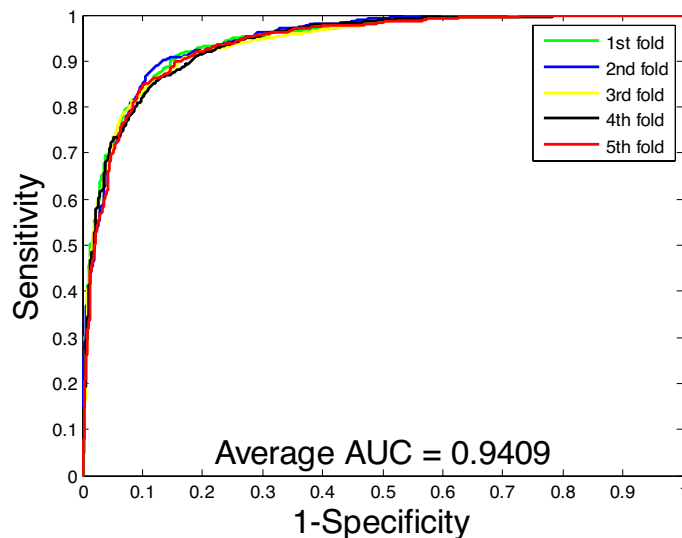
**Table 2.** Comparison of the results of the proposed method and SVM by using five-fold cross-validation on two datasets.

the parameters  $c$  and  $g$  are set to 4 and 1, respectively. When detecting PPIs on the *Human* dataset, the parameters  $c$  and  $g$  are set to 8 and 1, respectively. Furthermore, we chose the radial basis function as the kernel function in this experiment.

From Table 2, we can observe that the SVM-based method achieves an average accuracy of 78.96%, an average precision of 79.08%, an average sensitivity of 78.76%, and an average MCC of 66.80% by using fivefold cross-validation on the *Yeast* dataset. However, the RoF-based methods achieved average accuracy, precision, sensitivity,



**Figure 5.** ROC curves performed using the SVM method on *Yeast* dataset.



**Figure 6.** ROC curves performed using the SVM method on *Human* dataset.

and MCC of 90.07%, 90.24%, 89.83%, and 82.10%, respectively. At the same time, we also compared the prediction results of the two classifiers on the *Human* dataset using the same feature extraction method. Similarly, we can see that the SVM-based classifier has 87.23% average accuracy, 87.23% average precision, 85.83% average sensitivity, and 77.66% average MCC on the *Human* dataset. In addition, we plot the ROC curves on the two datasets based on the SVM model and calculate the average AUC as shown in Figs. 5 and 6. By comparing these experimental data, we can see that RoF classifiers are significantly better than SVM classifiers in predicting PPis.

**Comparison time performance with SVM-based method..** In this section, we compare the training time required by RoF and SVM algorithms on two datasets, by using the same OLPP feature extraction method on the same machine configuration. Table 3 gives the comparison results of the training time required by different algorithms on the *Yeast* and *Human* datasets. It can be shown that the training time of OLPP + RoF method is 401 s higher than that of OLPP + SVM method and the accuracy is improved by about 10% on the *Yeast* dataset. Similarly, the training time of OLPP + RoF method is 170 s while the training time of OLPP + SVM method is 110 s on the *Human* dataset. Although the training speed of the latter is 60 s faster than that of the former, the accuracy is reduced by about 9%. As a result, the RoF algorithm is superior to the SVM algorithm in terms of both prediction accuracy and training time.

Data sets	Method	Time (s)/fold	Accuracy (%)
Yeast	OLPP + RoF	237	90.21
	OLPP + SVM	638	79.53
Human	OLPP + RoF	170	96.69
	OLPP + SVM	110	87.25

**Table 3.** Comparison time performance with SVM-based methods.

Method	Feature	Classifier	Acc (%)	Sen (%)	Pre (%)	MCC (%)
Du et al. <sup>a</sup>	Mutiple	DL	94.43 ± 0.30	92.06 ± 0.36	96.65 ± 0.59	88.97 ± 0.62
Wong et al. <sup>b</sup>	PR-LPQ	RoF	93.92 ± 0.36	91.10 ± 0.31	96.45 ± 0.45	88.56 ± 0.63
Wang et al. <sup>c</sup>	Bio2Vec	CNN	93.30	92.70	93.55	87.49
You et al. <sup>d</sup>	MCD	SVM	91.36 ± 0.36	90.67 ± 0.69	91.94 ± 0.62	84.21 ± 0.59
An et al. <sup>e</sup>	PSSMMF	SVM	90.48 ± 0.76	90.26 ± 0.87	90.58 ± 0.98	82.84 ± 1.27
Wang et al. <sup>f</sup>	3-mers	CNN	90.26	88.14	91.65	82.38
Our method	OLPP	RoF	90.07 ± 0.60	89.83 ± 1.41	90.24 ± 0.56	82.10 ± 0.97
Guo et al. <sup>g</sup>	ACC	SVM	89.33 ± 2.67	89.93 ± 3.68	88.87 ± 6.16	N/A
Zhou et al. <sup>h</sup>	LD	SVM	88.56 ± 0.33	87.37 ± 0.22	89.50 ± 0.60	77.15 ± 0.68
Guo et al. <sup>i</sup>	AC	SVM	87.36 ± 1.38	87.30 ± 4.68	87.82 ± 4.33	N/A
You et al. <sup>j</sup>	Mutiple	PCA-EELM	87.00 ± 0.29	86.15 ± 0.43	87.59 ± 0.32	77.36 ± 0.44
Yang et al. <sup>k</sup>	LD	KNN	86.15 ± 1.17	81.03 ± 1.74	90.24 ± 1.34	N/A

**Table 4.** Performance comparisons of 12 methods on the *Yeast* dataset. N/A means not available. The values behind ± represent the standard deviation. <sup>a</sup>Mutiple: Multiple Features; DL: Deep Learning; results reported by Ref.<sup>31</sup>. <sup>b</sup>PR-LPQ: property response matrix-Local Phase Quantization; RoF: Rotation Forest; results reported by Ref.<sup>32</sup>. <sup>c</sup>Bio2Vec: bio-to-vector; CNN: convolution neural network; results reported by Ref.<sup>33</sup>. <sup>d</sup>MCD: Multi-scale Continuous and Discontinuous; SVM: Support Vector Machine; results reported by Ref.<sup>34</sup>. <sup>e</sup>PSSMMF: position-specific scoring matrix with multifeatures fusion; results reported by Ref.<sup>35</sup>. <sup>f</sup>3-mers: represent a segmentation strategy of sequence; results reported by Ref.<sup>33</sup>. <sup>g</sup>ACC: Auto Cross Covariance; results reported by Ref.<sup>23</sup>. <sup>h</sup>LD: Local Description; results reported by Ref.<sup>36</sup>. <sup>i</sup>AC: Auto Covariance; results reported by Ref.<sup>23</sup>. <sup>j</sup>PCA-EELM: Principal Component Analysis-Ensemble Extreme Learning Machine; results reported by Ref.<sup>25</sup>. <sup>k</sup>LD: Local descriptors; KNN: k-nearest neighbors; results reported by Ref.<sup>37</sup>.

**Comparison with other methods.** Thus far, many computational methods have been developed to detect PPIs. In particular, machine learning algorithms have also received widespread attention from researchers. In this section, we compare the proposed method with the currently known methods to further evaluate the predictive power of the model. Tables 4 and 5 summarize the predicted results of other existing methods on *Yeast* and *Human* datasets, respectively. From Table 4, we can see that the accuracy of the proposed method is 90.07%, the sensitivity is 89.83%, the precision is 90.24% and the MCC is 82.10% with the corresponding standard deviations of 0.60, 1.41, 0.56, and 0.97, respectively on the *Yeast* dataset. Similarly, we can find the prediction results of different methods on the *Human* dataset from Table 5. The average accuracy of the proposed method for PPIs prediction reached 96.09%, the sensitivity reached 95.20% and the MCC reached 92.47%. Comparing these results, we can find that the proposed method is a stable and reliable model for predicting PPIs.

**Performance on independent dataset.** Although the proposed model has achieved good performance on *Yeast* and *Human* datasets, the suitability of the proposed method for different datasets still needs to be verified. Therefore, we also performed additional experiments to further determine the predictive performance of this model for other species. It should be noticed that there is a biological hypothesis that PPIs are mapped from one species to another. This hypothesis is that many physically interacting proteins have coevolved in a given organism so that they are also likely to interact with proteins from other organisms. In this experiment, we used all of the 11,188 protein pairs of *Yeast* datasets to construct a training set through the previously proposed method. Then, we use four independent datasets as test sets to detect the final prediction model separately. Among them, the four independent test sets are *C. elegans*, *H. pylori*, *H. sapiens*, and *M. musculus* collected in the DIP database. The number of their protein pairs is 4013, 1420, 1412 and 313, respectively. Table 6 shows the PPIs prediction results of the five methods on four species. We can conclude that the proposed model achieved up to 90% prediction accuracy on four independent datasets *C. elegans*, *H. pylori*, *H. sapiens*, and *M. musculus*, which were 90.93%, 92.54%, 92.21%, and 91.37%, respectively. These results not only indicate the outstanding performance of the proposed method in predicting the interaction of other species but also show that the method has good generalization.



Method	Feature	Classifier	Acc (%)	Sen (%)	Pre (%)	MCC (%)
Du et al. <sup>a</sup>	Mutiple	DL	98.14	96.95	99.13	96.29
Ding et al. <sup>b</sup>	MMI + NMBAC	RF	97.56	96.57	98.30	95.13
Pan et al. <sup>c</sup>	LDA	RF	96.4	94.2	N/A	92.8
Huang et al. <sup>d</sup>	DTC + SMR	WSRC	96.30	92.63	99.59	92.82
Our method	OLPP	RoF	96.09	95.20	96.56	92.47
Ding et al. <sup>e</sup>	MMI	RF	96.08	95.05	96.97	92.17
Pan et al. <sup>f</sup>	LDA	RoF	95.7	97.6	N/A	91.8
Ding et al. <sup>g</sup>	NMBAC	RF	95.59	94.06	96.94	91.21
Pan et al. <sup>h</sup>	AC	RF	95.5	94.0	N/A	91.4
Pan et al. <sup>i</sup>	AC	RoF	95.1	93.3	N/A	91.0
Pan et al. <sup>j</sup>	LDA	SVM	90.7	89.7	N/A	81.3
Pan et al. <sup>k</sup>	AC	SVM	89.3	94.0	N/A	79.2

**Table 5.** Performance comparisons of 12 methods on the *Human* dataset. *N/A* means not available. The values behind  $\pm$  represent the standard deviation. <sup>a</sup>Mutiple: Multiple Features; DL: Deep Learning; results reported by Ref.<sup>31</sup>. <sup>b</sup>MMI + NMBAC: multivariate mutual information + normalized Moreau-Broto Autocorrelation; RF: Random Forest; results reported by Ref.<sup>38</sup>. <sup>c</sup>LDA: latent dirichlet allocation; RF: Random Forest; results reported by Ref.<sup>39</sup>. <sup>d</sup>DTC + SMR: discrete cosine transform + substitution matrix representation; WSRC: weighted sparse representation based classifier; results reported by Ref.<sup>19</sup>. <sup>e</sup>MMI: multivariate mutual information; results reported by Ref.<sup>38</sup>. <sup>f</sup>LDA: latent dirichlet allocation; RoF: Rotation Forest; results reported by Ref.<sup>39</sup>. <sup>g</sup>NMBAC: normalized Moreau-Broto Autocorrelation; results reported by Ref.<sup>38</sup>. <sup>h</sup>AC: auto covariance; results reported by Ref.<sup>39</sup>. <sup>i</sup>RoF: Rotation Forest; results reported by Ref.<sup>39</sup>. <sup>j</sup>SVM: Support Vector Machine; results reported by Ref.<sup>39</sup>. <sup>k</sup>Results reported by Ref.<sup>39</sup>.

Species	Test pairs	Our Method	Huang et al. <sup>19</sup>	Ding et al. <sup>40</sup>	Wang et al. <sup>41</sup>	Zhan et al. <sup>42</sup>
<i>C. elegans</i>	4013	90.93%	81.19%	86.72%	92.60%	93.20%
<i>H. pylori</i>	1420	92.54%	82.18%	90.34%	N/A	91.34%
<i>H. sapiens</i>	1412	92.21%	82.22%	90.23%	80.10%	91.93%
<i>M. musculus</i>	313	91.37%	79.87%	91.37%	89.14%	94.89%

**Table 6.** Performance comparisons on four species. *N/A* means not available.

## Conclusions

Machine learning algorithms play a crucial role in proteomics research as they can quickly and accurately improve the prediction accuracy of PPIs. In this work, we propose an ensemble learning approach to detect PPIs from protein sequences. Orthogonal locality preserving projections are used to extract discriminative features from the PSSM, which can effectively preserve evolutionary information of the protein sequence. Finally, we use a rotation forest model to predict PPIs. To evaluate the reliability of the proposed method for PPIs prediction, we performed experiments on *Yeast* and *Human* datasets to verify the performance of the method. At the same time, we also compared the proposed model with the SVM classifier and other existing models. The experimental results show that our method has achieved good performance in predicting protein interactions and it can be a useful tool for detecting PPIs.

## Materials and methodology

**Data sources.** Previous studies have generated many databases of protein–protein interactions, such as Bio-molecular Interaction Network Database (BIND)<sup>43</sup>, Molecular Interaction Database (MINT)<sup>44</sup>, and Database of Interacting Proteins (DIP)<sup>45</sup>. To demonstrate the efficacy of the proposed method, we used two publicly available and highly reliable datasets for this study, including *Yeast* and *Human*, which were derived from the database of interacting proteins (DIP) and collected by Guo et al.<sup>23</sup> and Huang et al.<sup>19</sup>, respectively. To eliminate the redundancy of the dataset and ensure the validity of the experiment, we performed a screening work to remove the redundant protein pairs<sup>46</sup>. Specifically, protein pairs with fewer than fifty residues are completely removed, as they may be just fragments. Furthermore, considering the presence of homologous sequence pairs, those protein pairs with more than 40% sequence identity were also removed. Finally, we retained the remaining 5594 protein pairs to construct a positive PPIs dataset. At the same time, we also constructed a negative dataset using an additional 5594 non-interacting protein pairs, and their subcellular localization was different. Thus, the final *Yeast* dataset in this experiment consisted of 11,188 protein pairs, which contained 50% negative datasets and 50% positive datasets. Analogously, we constructed 8161 protein pairs for *Human* dataset, which included 4262 non-interacting protein pairs and 3899 interacting protein pairs.

**Position-specific scoring matrix.** Gribskov et al.<sup>47</sup> initially introduced a position-specific scoring matrix (PSSM) for the search for distantly related proteins. PSSM is an evolutionary profile based on feature extraction methods that have been successfully used in various fields of bioinformatics. For instance, protein secondary structure prediction<sup>48</sup>, prediction of membrane protein types<sup>49</sup>, prediction of disordered regions<sup>50</sup>, identification of DNA binding proteins<sup>51</sup>, and protein binding site prediction<sup>52</sup>. To integrate the evolutionary information of proteins, we also used PSSM to predict PPIs in this study. The structure of the PSSM can be represented as a matrix with  $T$  rows and 20 columns. It can be interpreted as  $P = \{x_{ij} : i = 1, \dots, T, j = 1, \dots, 20\}$ . Of these, the rows of the matrix are protein residues and the columns refer to native amino acids. We can use the following formula to describe PSSM:

$$P = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,20} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,20} \\ \vdots & \vdots & \vdots & \vdots \\ x_{T,1} & x_{T,2} & \cdots & x_{T,20} \end{bmatrix}, \quad (5)$$

where  $T$  represents the length of the protein sequence, and the element  $x_{ij}$  of PSSM refers to the residue score of the  $i$ th residue mutated to the type  $j$  amino acid during biological evolution.

In this paper, we employed the Position-Specific Iterated BLAST (PSI-BLAST)<sup>53</sup> program and the SwissProt database on a local machine to transform each protein sequence into a matrix of score values to further construct experimental datasets to predict PPIs<sup>54</sup>. In the process of running PSI-BLAST, we hope to select highly homologous sequences, and mainly employ these aligned sequences to construct a new scoring matrix. This matrix is called the Position-Specific Scoring Matrix (PSSM), and is weighted according to the kinds of high homology found in the initial hit list. Using this matrix again, we do a blast to pick any new homologous sequences as our scoring schema will change. This process is repeated until no new sequences can be found. PSI-BLAST is more sensitive compared to BLAST, especially in terms of discovering new members of protein families. To generate highly homologous sequences, the important parameter cutoff  $e$ -value and the number of iterations of PSI-BLAST were set to 0.001 and 3, respectively, while other parameters were default values. The applications of PSI-BLAST can be publicly accessed at <http://blast.ncbi.nlm.nih.gov/Blast.cgi>.

**Orthogonal locality preserving projections (OLPP).** Orthogonal locality preserving projections (OLPP) algorithm is an effective manifold learning method. It was used early in the recognition of human faces and was proposed by Deng Cai et al.<sup>55</sup>. This algorithm is extended based on Locality preserving projections (LPP)<sup>56</sup>. Among them, the theoretical knowledge and detailed derivation of the LPP method can be traced back to Ref.<sup>57</sup>. Suppose we give a set of  $n$   $D$ -dimensional data  $x_1, x_2, \dots, x_n$  through  $n$   $d$ -dimensional vectors  $y_1, y_2, \dots, y_n$ , respectively,  $D > d$ . The objective function of LPP is formally stated below:

$$\min \sum_{ij} \|y_i - y_j\|^2 S_{ij}, \quad (6)$$

where  $S$  represents a similarity matrix and  $y_i$  is the one-dimensional representation of  $x_i$  with a projection vector  $w$ . Here,  $y_i = w^T x_i$ . According to the minimized objective function, LPP will incur a severe penalty if neighboring points  $x_i$  and  $x_j$  are projected far away. One possible way to define the similarity matrix  $S$  is as follows:

$$S_{ij} = \begin{cases} \exp\left(-\frac{1}{t} \|x_i - x_j\|^2\right), & \|x_i - x_j\|^2 < \varepsilon \\ 0, & \text{otherwise,} \end{cases} \quad (7)$$

where  $\varepsilon$  is extremely small,  $\varepsilon > 0$ , and the parameter  $t$  is seen as a regulator. Here,  $\varepsilon$  specifies the radius of the local neighborhood. That is,  $\varepsilon$  defines the locality. Thus, the objective function needs to be minimized so that when  $x_i$  and  $x_j$  are close,  $y_i$  and  $y_j$  are close as well. Finally, the transformation vector  $w$  is given by solving the minimum eigenvalue:

$$XLX^T w = \lambda XDX^T w, \quad (8)$$

where  $X = \{x_1, x_2, \dots, x_n\}$  and  $\lambda$  represents the eigenvalue and  $w$  is the corresponding eigenvector. Here,  $L = D - S$  is the Laplacian matrix and  $D$  represents a diagonal matrix,  $D_{ii} = \sum_j S_{ji}$ . Next, we describe the OLPP algorithm by using the following steps.

1. PCA projection. Principal Components Analysis (PCA) is an effective tool for reducing the dimensionality of multivariate data by using a covariance analysis between factors. PCA projects the input data into an alternate subspace by discarding the portion corresponding to zero eigenvalue. Here, we introduce the  $W_{PCA}$  to represent the transformation matrix of PCA.
2. Contiguity graph construction. OLPP algorithm can construct a K-nearest neighbor (KNN) graph in supervised or unsupervised mode and can also achieve good stability. Let  $G$  denote a KNN graph with  $n$  nodes. The  $i$ -th node corresponds to  $x_i$ . We tend to put an edge between nodes  $i$  and  $j$  if  $x_i$  and  $x_j$  are close, i.e.  $x_i$  is among  $p$  nearest neighbors of  $x_j$ . In other words,  $x_j$  is among  $p$  nearest neighbors of  $x_i$ . Edges are located between a sample and its  $K$  nearest neighbors in an unsupervised setting. Here,  $K$  represents a small integer. In general, we use the Euclidean distance metric to measure the closeness between data nodes in a  $K$  nearest neighbor



graph. In an unsupervised mode, we can get a constructed nearest neighbor graph that approximates the local manifold structure.

3. Selecting the weights. If node  $i$  and  $j$  are linked, the weight  $W_{ij}$  is expressed as,

$$W_{ij} = e^{-\frac{1}{t} \|x_i - x_j\|^2}, \quad (9)$$

where  $t$  is a suitable constant. If node  $i$  and  $j$  are not linked, we have  $W_{ij} = 0$ . The weight matrix  $W$  of graph  $G$  refers to the native structure of the feature space.

4. Computing the orthogonal basis functions. After finding the weight matrix  $W$ , we tend to calculate the diagonal matrix  $D$ . The diagonal matrix  $D$  is defined as the sums of each column element of  $W$  (or sums of each row element of  $W$  as  $W$  is symmetric):

$$D_{ii} = \sum_j W_{ji}. \quad (10)$$

We also calculated the Laplacian matrix  $L$ , which is defined as

$$L = D - W. \quad (11)$$

Let  $\{o_1, o_2, \dots, o_d\}$  be orthogonal basis vectors, and we define

$$A^{(d-1)} = [o_1, o_2, \dots, o_{d-1}], \quad (12)$$

$$B^{(d-1)} = [A^{(d-1)}]^T (XDX^T)^{-1} A^{(d-1)}. \quad (13)$$

The calculation process of the orthogonal basis vectors  $\{o_1, o_2, \dots, o_d\}$  can be expressed as follows

- (a) Compute  $o_1$  as the eigenvector of  $(XDX^T)^{-1} XDX^T$  associated with the smallest eigenvalue.
- (b) Compute  $o_d$  as the eigenvector of

$$M^{(d)} = \{I - (XDX^T)^{-1} A^{(d-1)} [B^{(d-1)}]^{-1} [A^{(d-1)}]^T\} \cdot (XDX^T)^{-1} XDX^T \quad (14)$$

related to the minimum eigenvalue of  $M^{(d)}$ .

5. OLPP embedding. Let  $W_{OLPP} = [o_1, o_2, \dots, o_s]$ , the embedding is defined as,

$$x \rightarrow y = W^T x, \quad (15)$$

$$W = W_{PCA} W_{OLPP}, \quad (16)$$

where  $y$  is a  $s$ -dimensional vector and  $W$  is the transformation matrix.

**Rotation forest.** In recent years, many ensemble algorithms have been rapidly developed in the field of machine learning, mainly because the ensemble learning classification method can greatly improve the prediction accuracy of classification results. Among them, ensemble classifier built using ensemble machine learning algorithms, such as boosting and bagging methods, usually have much better prediction accuracy than using only a single classifier. In this paper, we use the Rotation Forest (RoF) classifier to perform the classification task of protein–protein interactions. Rotation forest is an ensemble classifier combining decision tree algorithm and principal component analysis theory, which was proposed by Rodriguez et al.<sup>58</sup>. The main idea of the RoF classifier is to improve the diversity and prediction accuracy of the base classifiers by using a transformation approach to perform feature extractions for each classifier<sup>59</sup>. In addition, each decision tree is individually trained and embedded in a rotated feature space utilizing a new dataset in the transformed feature space by the original dataset<sup>60</sup>. Other research literature suggests that the RoF algorithm can achieve better prediction accuracy in classification problems when compared to other ensemble methods<sup>61,62</sup>.

Assuming  $X$  be the original training dataset and we can represent it with a matrix of  $N \times n$ . Here,  $N$  denotes the number of training samples and  $n$  denotes the number of features. The corresponding feature set and class label can be represented as  $S$  and  $Y$ , respectively, where  $Y = (y_1, y_2, \dots, y_n)^T$ . Let  $L$  be the total number of decision tree classifiers present in the RoF, where the  $i$ th decision tree is  $T_i$  ( $i = 1, 2, \dots, L$ ). More specifically, the feature set  $S$  is first randomly divided into  $K$  disjoint subsets in the rotation forest model. In each subset, there are  $C = \frac{n}{K}$  features. Here,  $K$  and  $L$  are two user-defined parameters. Next, we can get  $S_{ij}$  and  $X_{ij}$ , where  $S_{ij}$  is the  $j$ th subset of features for the  $i$ th decision tree classifier and  $X_{ij}$  is the training dataset  $X$  for features in  $S_{ij}$ . Based on the bootstrap algorithm, we can generate a new nonempty training set  $X'_{ij}$ , which is 75% of the original training dataset. Furthermore, a linear transformation method is applied to  $X'_{ij}$  to generate a coefficient vector, and it can be described as  $\{a_{ij}^{(1)}, \dots, a_{ij}^{(C_j)}\}$ , and the size of each  $X'_{ij}$  is  $C \times 1$ . Subsequently, a sparse rotation transformation matrix  $G_i$  can be constructed, as shown in the following equation:

$$G_i = \begin{bmatrix} a_{i1}^{(1)}, \dots, a_{i1}^{(C_1)} & \{0\} & \dots & \{0\} \\ \{0\} & a_{i2}^{(1)}, \dots, a_{i2}^{(C_2)} & \dots & \{0\} \\ \vdots & \vdots & \ddots & \vdots \\ \{0\} & \{0\} & \dots & a_{iK}^{(1)}, \dots, a_{iK}^{(C_K)} \end{bmatrix}. \quad (17)$$

Then, for a given test sample  $x$ , the  $d_{ij}(xG_i^a)$  generated by the decision tree classifier  $T_i$  is used to determine that the sample  $x$  belongs to the class  $y_i$ . In the next step, the average combination method is used for each class  $y_i$  to calculate the confidence and the formula is as follows:

$$m_j(x) = \frac{1}{L} \sum_{i=1}^L d_{ij}(xG_i^a). \quad (18)$$

Accordingly, for a given test sample  $x$ , the main purpose is to assign it to the class with the highest confidence. Thus, to determine whether these protein pairs have interactions with each other.

Received: 13 October 2020; Accepted: 15 April 2021

Published online: 19 August 2021

## References

- Zhang, Q. C. *et al.* Structure-based prediction of protein–protein interactions on a genome-wide scale. *Nature* **490**, 556 (2012).
- Ito, T. *et al.* A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl. Acad. Sci.* **98**, 4569–4574 (2001).
- Koegl, M. & Uetz, P. Improving yeast two-hybrid screening systems. *Brief. Funct. Genom. Proteomic.* **6**, 302–312 (2007).
- Zhu, H. & Snyder, M. Protein chip technology. *Curr. Opin. Chem. Biol.* **7**, 55–63 (2003).
- Puig, O. *et al.* The tandem affinity purification (TAP) method: A general procedure of protein complex purification. *Methods* **24**, 218–229 (2001).
- Niranjankumari, S., Lasda, E., Brazas, R. & Garcia-Blanco, M. A. Reversible cross-linking combined with immunoprecipitation to study RNA–protein interactions in vivo. *Methods* **26**, 182–190 (2002).
- Xenarios, I. *et al.* DIP, the database of interacting proteins: A research tool for studying cellular networks of protein interactions. *Nucleic Acids Res.* **30**, 303–305 (2002).
- Xia, J.-F., You, Z.-H., Wu, M., Wang, S.-L. & Zhao, X.-M. Improved method for predicting  $\pi$ -turns in proteins using a two-stage classifier. *Protein Pept. Lett.* **17**, 1117–1122 (2010).
- You, Z.-H., Li, L., Yu, H., Chen, S. & Wang, S.-L. Increasing Reliability of Protein Interactome by Combining Heterogeneous Data Sources with Weighted Network Topological Metrics. In: *International Conference on Intelligent Computing*. Springer. 657–663 (2010).
- Lei, Y.-K., You, Z.-H., Dong, T., Jiang, Y.-X. & Yang, J.-A. Increasing reliability of protein interactome by fast manifold embedding. *Pattern Recogn. Lett.* **34**, 372–379 (2013).
- Hamp, T. & Rost, B. More challenges for machine-learning protein interactions. *Bioinformatics* **31**, 1521–1525 (2015).
- Park, Y. Critical assessment of sequence-based protein–protein interaction prediction methods that do not require homologous protein sequences. *BMC Bioinform.* **10**, 1–13 (2009).
- Chiang, J.-H. & Yu, H.-C. Literature extraction of protein functions using sentence pattern mining. *IEEE Trans. Knowl. Data Eng.* **17**, 1088–1098 (2005).
- Enright, A. J., Iliopoulos, I., Kyripides, N. C. & Ouzounis, C. A. Protein interaction maps for complete genomes based on gene fusion events. *Nature* **402**, 86 (1999).
- Sun, J. *et al.* Refined phylogenetic profiles method for predicting protein–protein interactions. *Bioinformatics* **21**, 3409–3415 (2005).
- Mahdavi, M. A. & Lin, Y.-H. False positive reduction in protein–protein interaction predictions using gene ontology annotations. *BMC Bioinform.* **8**, 262 (2007).
- Göktepe, Y. E. & Kodaz, H. Prediction of protein–protein interactions using an effective sequence based combined method. *Neurocomputing* **303**, 68–74 (2018).
- Jothi, R., Cherukuri, P. F., Tasneem, A. & Przytycka, T. M. Co-evolutionary analysis of domains in interacting proteins reveals insights into domain–domain interactions mediating protein–protein interactions. *J. Mol. Biol.* **362**, 861–875 (2006).
- 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**, 902198. <https://doi.org/10.1155/2015/902198> (2015).
- Li, Y. *et al.* An ensemble classifier to predict protein–protein interactions by combining PSSM-based evolutionary information with local binary pattern model. *Int. J. Mol. Sci.* **20**, 3511 (2019).
- Yu, H.-J. & Huang, D.-S. Normalized feature vectors: A novel alignment-free sequence comparison method based on the numbers of adjacent amino acids. *IEEE ACM Trans. Comput. Biol. Bioinform. TCBB* **10**, 457–467 (2013).
- Luo, X. *et al.* A highly efficient approach to protein interactome mapping based on collaborative filtering framework. *Sci. Rep.* **5**, 7702 (2015).
- 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.* **36**, 3025–3030 (2008).
- Pitre, S. *et al.* PIPE: A protein–protein interaction prediction engine based on the re-occurring short polypeptide sequences between known interacting protein pairs. *BMC Bioinform.* **7**, 365 (2006).
- 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 Bioinform.* **14**, S10. <https://doi.org/10.1186/1471-2105-14-s8-s10> (2013).
- Lei, X. & Liang, J. Neighbor affinity-based core-attachment method to detect protein complexes in dynamic PPI networks. *Molecules* **22**, 1223 (2017).
- Ding, Y., Tang, J. & Guo, F. Identification of protein–protein interactions via a novel matrix-based sequence representation model with amino acid contact information. *Int. J. Mol. Sci.* **17**, 1623 (2016).
- Wang, Y. *et al.* PCVMZM: Using the probabilistic classification vector machines model combined with a zernike moments descriptor to predict protein–protein interactions from protein sequences. *Int. J. Mol. Sci.* **18**, 1029 (2017).
- Hanley, J. A. & McNeil, B. J. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* **143**, 29–36 (1982).

30. Huang, J. & Ling, C. X. Using AUC and accuracy in evaluating learning algorithms. *IEEE Trans. Knowl. Data Eng.* **17**, 299–310 (2005).
31. Du, X. *et al.* DeepPPI: Boosting prediction of protein–protein interactions with deep neural networks. *J. Chem. Inf. Model.* **57**, 1499–1510 (2017).
32. 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. In: *International Conference on Intelligent Computing*. Springer. 713–720 (2015).
33. Wang, Y. *et al.* A high efficient biological language model for predicting protein–protein interactions. *Cells* **8**, 122 (2019).
34. You, Z.-H. *et al.* Prediction of protein–protein interactions from amino acid sequences using a novel multi-scale continuous and discontinuous feature set. *BMC Bioinformatics*. **15**, 1–9 (2014).
35. An, J.-Y., Zhou, Y., Zhao, Y.-J. & Yan, Z.-J. An efficient feature extraction technique based on local coding PSSM and multifeatures fusion for predicting protein–protein interactions. *Evol. Bioinform.* **15**, 1176934319879920 (2019).
36. Zhou, Y. Z., Gao, Y. & Zheng, Y. Y. Prediction of protein–protein interactions using local description of amino acid sequence. In: *Advances in Computer Science and Education Applications*. Communications in Computer and Information Science, vol. 202 (eds Zhou, M. & Tan, H.) 254–262 (Springer, Berlin, Heidelberg, 2011).
37. Yang, L., Xia, J.-F. & Gui, J. Prediction of protein–protein interactions from protein sequence using local descriptors. *Protein Pept. Lett.* **17**, 1085–1090 (2010).
38. Ding, Y., Tang, J. & Guo, F. Predicting protein–protein interactions via multivariate mutual information of protein sequences. *BMC Bioinform.* **17**, 1–13 (2016).
39. Pan, X.-Y., Zhang, Y.-N. & Shen, H.-B. Large-scale prediction of human protein–protein interactions from amino acid sequence based on latent topic features. *J. Proteome Res.* **9**, 4992–5001 (2010).
40. Ding, Y., Tang, J. & Guo, F. Predicting protein–protein interactions via multivariate mutual information of protein sequences. *BMC Bioinform.* **17**, 398 (2016).
41. Wang, Y.-B., You, Z.-H., Li, L.-P., Huang, Y.-A. & Yi, H.-C. Detection of interactions between proteins by using legendre moments descriptor to extract discriminatory information embedded in pssm. *Molecules* **22**, 1366 (2017).
42. Zhan, X.-K. *et al.* Using random forest model combined with gabor feature to predict protein–protein interaction from protein sequence. *Evol. Bioinform.* **16**, 1176934320934498 (2020).
43. Bader, G. D., Betel, D. & Hogue, C. W. BIND: The biomolecular interaction network database. *Nucleic Acids Res.* **31**, 248–250 (2003).
44. Licata, L. *et al.* MINT, the molecular interaction database: 2012 update. *Nucleic Acids Res.* **40**, D857–D861 (2011).
45. Salwinski, L. *et al.* The database of interacting proteins: 2004 update. *Nucleic Acids Res.* **32**, D449–D451 (2004).
46. Aloy, P., Ceulemans, H., Stark, A. & Russell, R. B. The relationship between sequence and interaction divergence in proteins. *J. Mol. Biol.* **332**, 989–998 (2003).
47. Gribskov, M., McLachlan, A. D. & Eisenberg, D. Profile analysis: Detection of distantly related proteins. *Proc. Natl. Acad. Sci.* **84**, 4355–4358 (1987).
48. Jones, D. T. Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.* **292**, 195–202 (1999).
49. Pu, X., Guo, J., Leung, H. & Lin, Y. Prediction of membrane protein types from sequences and position-specific scoring matrices. *J. Theor. Biol.* **247**, 259–265 (2007).
50. Jones, D. T. & Ward, J. J. Prediction of disordered regions in proteins from position specific score matrices. *Proteins Struct. Funct. Bioinform.* **53**, 573–578 (2003).
51. Waris, M., Ahmad, K., Kabir, M. & Hayat, M. Identification of DNA binding proteins using evolutionary profiles position specific scoring matrix. *Neurocomputing* **199**, 154–162 (2016).
52. Chen, X.-W. & Jeong, J. C. Sequence-based prediction of protein interaction sites with an integrative method. *Bioinformatics* **25**, 585–591 (2009).
53. Altschul, S. F. *et al.* Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res.* **25**, 3389–3402 (1997).
54. Shen, H. & Chou, J. J. MemBrain: Improving the accuracy of predicting transmembrane helices. *PLoS ONE* **3**, e2399 (2008).
55. Cai, D. & He, X. Orthogonal locality preserving indexing. In: *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3–10 (2005).
56. He, X., Yan, S., Hu, Y., Niyogi, P. & Zhang, H. J. Face recognition using laplacianfaces. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**, 328–340. <https://doi.org/10.1109/tpami.2005.55> (2005).
57. He, X. & Niyogi, P. Locality preserving projections. *Adv. Neural Inf. Process. Syst.* **16**, 153–160 (2004).
58. Rodriguez, J. J., Kuncheva, L. I. & Alonso, C. J. Rotation forest: A new classifier ensemble method. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**, 1619–1630 (2006).
59. Kotsiantis, S. Combining bagging, boosting, rotation forest and random subspace methods. *Artif. Intell. Rev.* **35**, 223–240 (2011).
60. Xia, J.-F., Han, K. & Huang, D.-S. Sequence-based prediction of protein–protein interactions by means of rotation forest and autocorrelation descriptor. *Protein Pept. Lett.* **17**, 137–145 (2010).
61. Liu, K.-H. & Huang, D.-S. Cancer classification using rotation forest. *Comput. Biol. Med.* **38**, 601–610 (2008).
62. Du, P., Samat, A., Waske, B., Liu, S. & Li, Z. Random forest and rotation forest for fully polarized SAR image classification using polarimetric and spatial features. *ISPRS J. Photogramm. Remote Sens.* **105**, 38–53 (2015).

## Acknowledgements

This work is supported in part by the National Science Foundation of China, under Grant 61873212, 61722212, 61902342 and 62072378. The authors would like to thank all the editors and anonymous reviewers for their constructive advices.

## Author contributions

Y.L., Z.W. and L.-P.L. conceived the algorithm, carried out experiments, and wrote the manuscript. Z.-H.Y., W.-Z.H., X.-K.Z. and Y.-B.W. prepared the data sets, designed, performed and analyzed experiments.

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to Z.W. or L.-P.L.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2021