Computers in Biology and Medicine 123 (2020) 103899

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: http://www.elsevier.com/locate/compbiomed

Improving protein-protein interactions prediction accuracy using XGBoost feature selection and stacked ensemble classifier

Cheng Chen [a,b,1], Qingmei Zhang [a,b,1], Bin Yu [a,b,c,*,1], Zhaomin Yu [a,b], Patrick J. Lawrence [d], Qin Ma [d], Yan Zhang [e]

[a] College of Mathematics and Physics, Qingdao University of Science and Technology, Qingdao, 266061, China
[b] Artificial Intelligence and Biomedical Big Data Research Center, Qingdao University of Science and Technology, Qingdao, 266061, China
[c] School of Life Sciences, University of Science and Technology of China, Hefei, 230027, China
[d] Department of Biomedical Informatics, College of Medicine, The Ohio State University, Columbus, OH, 43210, USA
[e] College of Electromechanical Engineering, Qingdao University of Science and Technology, Qingdao, 266061, China

ARTICLE INFO

Keywords:
Protein-protein interactions
Multi-information fusion
XGBoost
Stacked ensemble classifier

ABSTRACT

Protein-protein interactions (PPIs) are involved with most cellular activities at the proteomic level, making the study of PPIs necessary to comprehending any biological process. Machine learning approaches have been explored, leading to more accurate and generalized PPIs predictions. In this paper, we propose a predictive framework called StackPPI. First, we use pseudo amino acid composition, Moreau-Broto, Moran and Geary autocorrelation descriptor, amino acid composition position-specific scoring matrix, Bi-gram position-specific scoring matrix and composition, transition and distribution to encode biologically relevant features. Secondly, we employ XGBoost to reduce feature noise and perform dimensionality reduction through gradient boosting and average gain. Finally, the optimized features that result are analyzed by StackPPI, a PPIs predictor we have developed from a stacked ensemble classifier consisting of random forest, extremely randomized trees and logistic regression algorithms. Five-fold cross-validation shows StackPPI can successfully predict PPIs with an ACC of 89.27%, MCC of 0.7859, AUC of 0.9561 on *Helicobacter pylori*, and with an ACC of 94.64%, MCC of 0.8934, AUC of 0.9810 on *Saccharomyces cerevisiae*. We find StackPPI improves protein interaction prediction accuracy on independent test sets compared to the state-of-the-art models. Finally, we highlight StackPPI's ability to infer biologically significant PPI networks. StackPPI's accurate prediction of functional pathways make it the logical choice for studying the underlying mechanism of PPIs, especially as it applies to drug design. The datasets and source code used to create StackPPI are available here: https://github.com/QUST-AIBBDRC/StackPPI/.

1. Introduction

Studying protein-protein interactions (PPIs) can help elucidate the mechanisms of most cellular processes at the proteomic level [1,2]. Computational modeling using sequence, evolution and structure based data has been employed to address current experimental limitations, and in doing so, attempts to identify, validate and improve the prediction of PPIs [3]. Machine-learning (ML) based methods have proven especially successfully at inferring PPIs from provided protein pairs. Such advancements mitigate the impact of experimental shortcomings on PPIs prediction while simultaneously bolstering our understanding of protein interactions from a computational perspective. Improved computational methods can reveal the etiology and pathogenesis of diseases, as well as, provide greater insight into canonical pathways through the construction of protein interaction networks [4–7].

Predicting PPIs is, in essence, a binary classification task, requiring a provided protein pair to be labeled as either 'interacting' or 'non-interacting'. The primary work of this task is how to unify the length of input biological sequences [8–10]. At present, various feature encoding schemes are merged, containing sequence-, network-, evolution-, position- and structure based approaches [11–14]. Wang et al. [15], for example, employed a linear programming algorithm to assess protein structure data. This framework identified that utilizing features from cooperative protein domains improved their PPI predictions. Taking a

* Corresponding author. College of Mathematics and Physics, Qingdao University of Science and Technology, Qingdao, 266061, China.
E-mail address: yubin@qust.edu.cn (B. Yu).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.compbiomed.2020.103899
Received 23 January 2020; Received in revised form 28 June 2020; Accepted 28 June 2020
Available online 15 July 2020

different approach, Hamp et al. [16] designed an evolutionary profile kernel-based support vector machine (SVM) sequence predictor, and the input features were represented by k-mer. By filtering by gene expression Hamp et al.'s evolution-based method successfully improved PPIs predictions. Considering the effect of local peptide and whole sequence, You et al. [17] used multi-scale local descriptor (MLD) combined with random forest (RF) for predicting PPIs. Sequence-based schemes with a powerful discrimination ability does not need prior knowledge, which is advantageous for experimental validation.

In recent years, progress has been made using sequence-based predictive method for PPIs. An et al. [18] proposed an effective and simple method, RVM-AB. The relevant vector machine (RVM) used average blocks algorithm (AB) to obtain the relationship between PPIs annotation and sequence evolutionary information. Specially, the RVM-AB model consistently predicted protein interactions from a *Saccharomyces cerevisiae* (*S. cerevisiae*) dataset, with greater accuracy than SVM-based models. With protein interaction data obtained from *Helicobacter pylori* (*H. pylori*), RVM-AB also outperformed other baseline tools. Zhang et al. [19] utilized NIP-SS and NIP-RW to construct non-interacting datasets (negative examples) to enable further study of PPIs. The non-interacting protein pairs produced by NIP-SS and NIP-RW enable the development of more robust ML-based models. Wang et al. [20] automated PPIs detection with a DNN-LCTD approach, which captured continuous and discontinuous, local and whole features using LCTD. The deep neural network (DNN) classifier can then extract additional information from the raw features produced by LCTD. Zhang et al. [21] combined both ensemble and deep learning approaches in EnsDNN to predict PPIs. Their model consisted of nine diverse DNNs with various configurations. Two hidden-layer NN created an ensemble of the outputs of 27 DNNs. The scheme proved quite accurate and robust when tested with gold standard PPIs datasets. Additionally, Ding et al. [22] proposed a new matrix-based representation method (amino acid contact matrix) combined with RF to classify protein pairs as interacting or not.

These advances demonstrate the promise of using ML models to inform PPI predictions, yet the current methodologies can be further developed. We have identified multiple areas in which the current state-of-the-art models can be improved. First, the overwhelming majority of models found in the literature only use information extracted from singular features. Such attempts are insufficient and fail to generalize appropriately. This results in a reduction in prediction accuracy on test data and in turn, an inability to ascertain any significant biological insights. How to effectively fuse multiple biological information is a problem. Additionally, researchers often use single classifiers to infer PPI predictions. Constructing such models consistently fail to effectively predict PPIs due to poor generalizability. The integration of multiple, unique classifiers enhances predictive performance [23]. Recent work suggests models which use stacked ensemble classifiers perform especially well on training and independent testing data [24–26]. Finally, the emergence of PPIs databases necessitates the development of a quick and accurate algorithm that can learn from and capitalize on the ever increasing availability of experimentally validated PPIs sequence data.

We constructed a computational PPIs prediction model, StackPPI, to address the shortcomings of state-of-the-art methodologies. We began by encoding biological feature vectors from *H. pylori* and *S. cerevisiae* datasets into pseudo amino acid composition (PAAC), Moreau-Broto, Moran and Geary autocorrelation descriptor (AD), amino acid composition position specific scoring matrix (AAC-PSSM), Bi-gram position specific scoring matrix (Bi-PSSM), composition, transition and distribution (CTD), which were all subsequently fused. StackPPI's performance was enhanced by the noise elimination and dimensionality reduction afforded by XGBoost. The utilization of XGBoost for feature selection is novel to PPIs prediction models. Another novel aspect of StackPPI is the construction of its stacked ensemble classifier which employs RF and extremely randomized trees (ET) as the base-classifiers, and the logistic regression (LR) as the meta-classifier. Optimized biological sequence feature vectors derived via XGBoost are provided as input into the stacked ensemble classifier. This enables our StackPPI model to learn essential features representing PPIs via two-layered learning. Model evaluations indicates StackPPI improves PPIs prediction accuracies over state-of-the-art predictors. Finally, we harnessed StackPPI to produce network visualizations for two PPI networks including the Wnt-related pathway and a disease-specific to further evaluate StackPPI's capabilites.

2. Materials and methods

2.1. PPIs datasets

We obtained eight PPI datasets, two of which were used to train our model and the remaining six served as benchmarks for evaluating StackPPI's performance. The first training set, constructed by Martin et al. [27] consists of 1458 positive samples (interacting protein pairs), and 1458 negative samples (non-interacting protein pairs) present in *H. pylori*. The negative examples are comprised of protein pairs not identified during experimentation. The second, obtained from the DIP database [28], includes 5594 interacting and 5594 non-interacting protein pairs [29]. For the second training dataset, protein pairs were deleted if the sequence length was less than fifty amino acid residues long, or the similarity between sequences were greater than or equal to 40%. Our independent test datasets include 1412 interacting pairs identified in *Homo sapiens* (*H. sapiens*), 313 interacting pairs identified in *Mus. Musculus* (*M. musculus*), 4013 interacting pairs identified in *Caenorhabditis elegans* (*C. elegans*) and 6954 interacting pairs identified in *Escherichia coli* (*E. coli*) [30]. Protein pairs comprising the Wnt-related pathway (96 PPIs pairs) [31] and the disease-specific (108 PPIs pairs) [1] were input into StackPPI to assess our model's capacity for inferring and predicting protein interaction networks.

2.2. Feature extraction methods

2.2.1. Pseudo amino acid composition

PAAC was first introduced in 2001 by Chou et al. [32] to encode and extract numerical vectors from raw amino acid sequence data and continues to be utilized in efforts to advance the fields genomics and proteomics [33–38]. PAAC characterizes and represents the composition frequency of each amino acid and incorporates sequence based data into its pseudo components [39–41].

$$p_u = \begin{cases} \dfrac{f_\mu}{\sum_{\mu=1}^{20} f_\mu + \omega \sum_{k=1}^{\lambda} \tau_k}, 1 \le \mu \le 20 \\ \dfrac{\omega \tau_{\mu-20}}{\sum_{\mu=1}^{20} f_\mu + \omega \sum_{k=1}^{\lambda} \tau_k}, 21 \le \mu \le 20+\lambda \end{cases} \quad (1)$$

where $\omega = 0.05$, f_μ is the frequency of the amino acid. τ_k is $k-th$ sequence correlation factor.

2.2.2. Morean-Broto, Moran, and Geary autocorrelation descriptor

Autocorrelation descriptors (AD) fall into three categories: Morean-Broto, Moran, and Geary autocorrelation, which defined by equations (2)–(4) respectively [42]. AD are used to identify non-randomness in the amino acid sequence. Amino acid residues are replaced with numerical signals for seven distinct physicochemical properties, generating a $3 \times 7 \times m$ dimensional feature vector to express physicochemical property-based and sequence-order information. We expound upon this process in the Supplementary Material Table S1.

$$NMBA(l) = \frac{NBA(l)}{N-l}, \; l = 1, 2, \cdots, m \quad (2)$$

$$MA(l)=\frac{\frac{1}{N-l}\sum_{i=1}^{N-l}(P(AA_i)-\overline{P})(P(AA_{i+l})-\overline{P})}{mean},\ l=1,2,\cdots,m \tag{3}$$

$$GA\left(l\right)=\frac{\frac{1}{2(N-l)}\sum_{i=1}^{N-l}PAA_i)-P(AA_{i+l}))^2}{mean},\ l=1,2,\cdots,m \tag{4}$$

where $MBA(l)=\sum_{i=1}^{N-l}P(AA_i)P(AA_{i+l})$, $P(AA_i)$ and $P(AA_{i+l})$ are physicochemical values, m is the interval need to be adjusted, $mean=\frac{1}{N}\sum_{i=1}^{N}(P(AA_i)-\overline{P})^2$, $\overline{p}$ represents average value.

2.2.3. Evolutionary information

ACC-PSSM and Bi-PSSM, equations (5) and (6), respectively, extract and integrate evolution and sequence based characteristics [43]. The position specific scoring matrix for both could be generated through PSI-BLAST [44]. Using their defined equations, the dimension of extracted feature vectors from AAC-PSSM and Bi-PSSM were found to be 20 and 400, respectively.

$$P_{AAC}=\left(p_1,p_2,\cdots,p_j,\cdots,p_{20}\right)^T\ (j=1,2,\cdots,20) \tag{5}$$

where $P_j=\frac{1}{L}\sum_{i=1}^{L}p_{ij}\ (j=1,2,\cdots 20)$ and p_j is the average score, representing the PSSM-based composition information of the $j-th$ column of PSSM.

Bi-PSSM calculates the transition from $m-th$ row to $n-th$ column of PSSM.

$$P_{Bi}=[B_{1,1},B_{1,2},\cdots,B_{1,20},B_{2,1},\cdots,B_{2,20},\cdots,B_{20,1},\cdots,B_{20,20}]^T \tag{6}$$

where $B_{m,n}=\sum_{i=1}^{L-1}p_{i,m}p_{(i+1),n}$, $p_{i,m}$ is the score of the PSSM.

2.2.4. Sequence information

CTD [21,42,45] is employed by our model to characterize the distribution pattern and 13 physicochemical information of protein sequences. Amino acids are grouped using 13 physicochemical properties. The detailed groups are shown in Table S2. For example, 'MQRPG PRLWLVLQVMGSCAAISSMDMERP' can be replaced as 'HPPNNNP HHHHHPHHNNHNNHNNHPHPPN' based on hydrophobicity.

The calculation process of composition descriptor can generate three values:

$$Composition(r)=\frac{L(r)}{L},\ r\in\{P,\ N,\ H\} \tag{7}$$

where $L(r)$ is the frequency of r, and there are twelve 'H', seven 'P', ten 'N'. L is the length of the protein sequence, which is twenty-nine. So the three values are $12/29=0.4138$, $7/29=0.2414$, $10/29=0.3448$ respectively.

The transition descriptor converts the protein sequence into the replaced sequence based on the grouped situation. Transition represents the dipeptide frequency values on the replaced sequence:

$$Transition(r,s)=\frac{L(r,s)+L(s,r)}{L-1},\ r,s\in\{(P,N),\ (N,H),(H,P)\} \tag{8}$$

where $L(r,s)$ is the frequency of rs, the percentage frequency of 'H' to 'P' or 'P' to 'H' is $7/28=0.25$. We also calculate the frequency of 'N' to 'P' or 'P' to 'N' is $3/28=0.1071$, and the value of 'N' to 'H' or 'H' to 'N' is $5/28=0.1786$.

The distribution descriptor includes five features for each group (polar, neutral and hydrophobic). For each given group, the first, 25%, 50%, 75%, and 100% residue position is divided by L.

For each protein sequence example, the dimension of CTD is 273 (The dimensions of C,T, D are 39, 39, 195, respectively).

2.3. XGBoost feature selection

XGBoost [46], a gradient boosting decision tree, uses regularized learning and cache-aware block structure tree learning for ensemble learning. L represents the loss function; f_t represents the $t-th$ tree and $\Omega(f_t)$ is regularized term. The second-order Taylor series of L at the $t-th$ iteration is:

$$L^{(t)}\simeq\sum_{i=1}^{k}\left[l\left(y_i,y_i^{(t-1)}\right)+g_if_t(x_i)+\frac{1}{2}h_if_t^2(x)\right]+\Omega(f_t) \tag{9}$$

where $g_i,\ h_i$ denotes the first and second order gradients. During our training of XGBoost, we uses gain to determine the optimal split node.

$$gain=\frac{1}{2}\left[\frac{\left(\sum_{i\in I_L}g_i\right)^2}{\sum_{i\in I_L}h_i+\lambda}+\frac{\left(\sum_{i\in I_R}g_i\right)^2}{\sum_{i\in I_R}h_i+\lambda}-\frac{\left(\sum_{i\in I}g_i\right)^2}{\sum_{i\in I}h_i+\lambda}\right]-\gamma \tag{10}$$

where I_L and I_R represent samples of the left and right nodes after the segmentation, respectively. $I=I_L\cup I_R$. $\lambda,\ \gamma$ are the penalty parameters. Gain represents the gain score for each split of a tree, and the final feature importance score is calculated by the average gain. The average gain is the total gain of all trees divided by the total number of splits for each feature. The higher the feature importance score of XGBoost is, the more important and effective the corresponding feature is. We obtain the top-ranked features based on descending order of feature importance to characterize the PPIs. XGBoost feature selection method has been used in the area of bioinformatics [47–49], and it achieves good performance. The number of boosting tree is set as 500, the max depth is 15, the loss function is binary: logistic, and the others use default parameters. The XGBoost package can be downloaded at https://github.com/dmlc/xgboost.

2.4. Stacked ensemble classifier

The stacked ensemble classifier combines multiple individual learners with the intent of minimizing generalization error. A precedent of using stacked ensemble classifiers in bioinformatics research was established through previous work improving bacterial secreted effector identification, DNA-binding protein detection and MicroRNA Biomarkers prediction [24–26]. The stacked ensemble classifier algorithm conducts two-staged learning; the first stage employs a multiple classifier system and the second stage uses a meta-classifier. The encoded sequence, evolution, and physiochemical property derived features, representing protein pairs, are supplied into the first stage as numerical vectors and category labels. This produces the probabilistic data that is used as input by the meta-classifier which then labels protein pairs as either 'interacting' or 'non-interacting'.

For StackPPI, we elected to build our model with two RFs [50] and two extremely randomized trees (ET) [51] as our base classifiers. RF and ET are unique tree-based classifiers that, when used in cooperation, improves learning during the first stage. Our RF and ET algorithms all contain 500 decision trees. The Gini coefficient is to determine the optimal segmentation node for the RF algorithms. All other parameters for the RF and ET algorithms remain at their default values. The LR algorithm [52] is employed as the meta classifier in the second level of our stacked ensemble classifier. None of the parameters for the LR algorithm are altered from their default values. The RF, ET and LR algorithms were implemented from the Scikit-learn package [53]. The code for Algorithm 1 can be downloaded from https://github.com/QUST-AIBBDRC/StackPPI/. Algorithm 1's pseudocode is given below:

Algorithm 1: Stacked ensemble classifier

Input: Dataset $D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)\}$;

Base-classifier $f_1=RF_1, f_2=RF_2, f_3=ET_1, f_4=ET_2$

Meta-classifier $f=LR$

1: **for** $t=1,\cdots,4$:

2: Train the base-classifiers in the first stage

$h_t=f_t(D)$;

3: **end**

5: **for** $i=1,2,\cdots,m$:

6: **for** $t=1,\cdots,4$:

7: Generate new feature vector for each sample

$z_{it}=h_t(x_i)$;

8: **end**

9: $D'=D'\cup((z_{i1},\cdots,z_{i4}),y_i)$;

10: **end**

11: Train the meta-classifier in the second stage

$h'=f(D')$;

Output: $H(x)=h'(h_1(x),\cdots,h_4(x))$

2.5. Model construction and evaluation

The workflow used to construct StackPPI is illustrated in Fig. 1. This was implemented using Python 3.6 with 32.0 GB of RAM and a Windows operating system.

The input of StackPPI should be protein sequence pairs. Then PAAC, AD, AAC-PSSM, Bi-PSSM and CTD are integrated. Then we adopt gain in XGBoost to select features with high score. Finally, RF, ET and LR algorithms are utilized to construct StackPPI model. The output are the probabilities to determine protein pair 'interacting' or 'non-interacting'.

We use five-fold cross-validation to perform the parameter optimization and model evaluation. Five-fold cross-validation is to randomly divide the samples into five parts, four parts are employed as the training set, and the remaining one are employed as the test set. The average result of the five parts are employed as the final prediction result.

Step 1: Collect eight benchmark PPIs datasets from DIP database and previous literatures.

Step 2: Determine λ of PAAC in equation (1) and m of AD in equations (2)–(4) according to results of various parameters of line chart and tables through five-fold cross-validation. Then, PAAC, AD, AAC-PSSM, Bi-PSSM, and CTD can be integrated to encode fully features for given protein pairs, which could extract complementary and representative information compared with single feature encoding approach.

Step 3: XGBoost is utilized to reduce noisy features, maintain the significant raw features, and prevent overfitting via average gain. We also use ACC, MCC, ROC and PR curve to assess the performance of XGBoost with KPCA, SVD, MDS and LLE.

Step 4: Random forest and extremely randomized trees are stacked to construct base-classifiers, and logistic regression is employed to build up the meta-classifier. The effective representation of protein pairs after dimensionality reduction are input into stacked ensemble classifier, and StackPPI is constructed.

Step 5: We apply feature encoding, feature integration and feature selection on independent test sets and PPIs graphs datasets. Then we train StackPPI model on *S. cerevisiae* to perform cross-species prediction, and networks visualization.

The evaluation metrics contain Accuracy (ACC), Matthew's correlation coefficient (MCC), Sensitivity (SE), Precision (PRE), Specificity (SP) [54–56].

$$SE=1-\frac{N_-^+}{N^+} \tag{11}$$

$$SP=1-\frac{N_+^-}{N^-} \tag{12}$$

$$PRE=\frac{N^+-N_-^+}{N^+-N_-^++N_+^-} \tag{13}$$

$$ACC=1-\frac{N_-^++N_+^-}{N^++N^-} \tag{14}$$

$$MCC=\frac{1-\frac{N_-^++N_+^-}{N^++N^-}}{\sqrt{\left(1+\frac{N_+^--N_-^+}{N^+}\right)\left(1+\frac{N_-^+-N_+^-}{N^-}\right)}} \tag{15}$$

where N^+ is the number of interacting pairs, N^- is the number of non-interacting pairs, N_-^+ is the number of interacting pairs which are regarded as negative samples incorrectly. N_+^- is the number of non-interacting pairs which are predicted positive samples incorrectly. The ROC curve, PR curve, AUC, AUPR are also the significant indicators to evaluate StackPPI model [57,58].

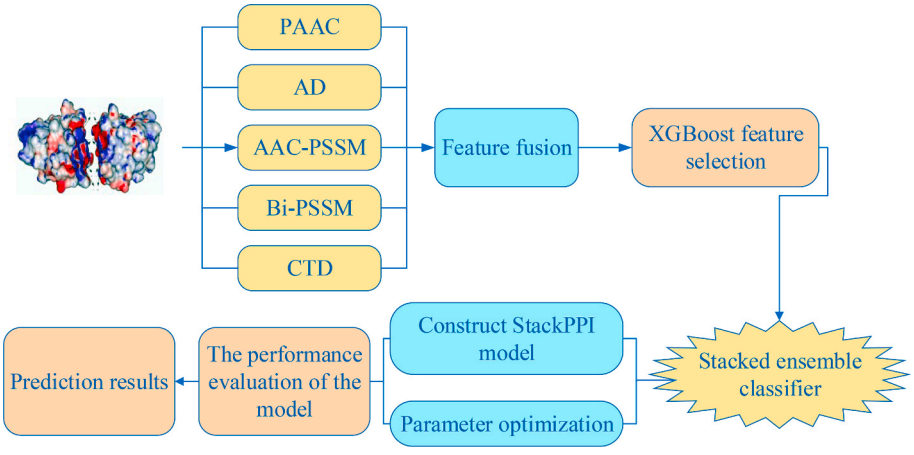

Fig. 1. The framework of StackPPI for PPIs prediction.

3. Results and discussion

3.1. The parameter optimization of λ and m

The PAAC and AD can obtain the sequence-based, position-based and physicochemical property-based features. From equations (1)–(4), the parameters λ in PAAC and m in AD need to be optimized. The shortest length of sequence in *H. pylori*, *S. cerevisiae* is 12. So λ and m are set as $1, 2, \cdots, 11$, and the proposed stacked ensemble classifier is employed to obtain accuracies under different parameters (Fig. 2), and the detailed results of parameter optimization are shown in Supplementary Table S3 and Table S4.

From Fig. 2 (A), we can see ACC values of the two datasets are different with the change of parameter λ values. When $\lambda = 11$, StackPPI's ACC reaches a global maximum using the *H. pylori* dataset, while that maximum is attained using the *S. cerevisiae* dataset when $\lambda = 9$. We use average accuracy to obtain the optimal parameter λ of PAAC in StackPPI ($\lambda = 11$). So the extracted dimension of PAAC is $20 + 11 = 31$. Fig. 2 (B) plots the ACC change of Moreau-Broto, Moran and Geary autocorrelation descriptor with different m. When $m = 8$, ACC of *H. pylori* reaches the highest, and when $m = 9$, StackPPI achieves the highest ACC (*S. cerevisiae*). We set $m = 9$ in StackPPI via the average prediction accuracy, and dimension of AD is $21 \times 9 = 189$.

3.2. Performance of different feature extraction methods

The raw dimensions of PAAC and AD are 31 and 189, respectively. The raw dimension of ACC-PSSM is 20, the raw dimension of Bi-PSSM is 400, and the raw dimension of CTD is 273. PAAC, AD, ACC-PSSM, Bi-PSSM and CTD are fused to generate 913-dimensional feature vectors. The protein pairs are concatenated to obtain 1826-dimensional feature vectors. PAAC and AD characterize the sequence-based, physicochemical property-based features, ACC-PSSM and Bi-PSSM characterize the evolution-based features, CTD characterizes the sequence-based and composition-based features. Integrate multiple biological features can fully illustrate PPIs and improve the generalization ability, some redundant and noisy features are produced. The XGBoost is used to reduce dimension. All-XGBoost represents the multi-information fusion and uses XGBoost feature selection to implement dimensionality reduction (the optimal feature subset number is 300). The prediction results of PAAC, AD, AAC-PSSM, Bi-PSSM, CTD and All-XGBoost are listed in Table 1.

Table 1
Comparison of feature fusion and single feature information on *H. pylori* and *S. cerevisiae* datasets.

Dataset	Algorithm	ACC (%)	PRE (%)	SE (%)	SP (%)	MCC
H. pylori	PAAC	87.89	89.57	85.80	89.99	0.7587
	AD	86.73	88.81	84.09	89.37	0.7358
	AAC-PSSM	84.94	85.83	83.74	86.15	0.6994
	Bi-PSSM	79.15	79.99	77.71	80.59	0.5837
	CTD	84.12	84.87	83.06	85.18	0.6827
	All-XGBoost	**89.27**	90.37	87.93	90.60	0.7859
S. cerevisiae	PAAC	94.43	96.06	92.67	96.16	0.8892
	AD	94.07	96.15	91.83	96.32	0.8824
	AAC-PSSM	93.71	94.93	92.35	95.07	0.8745
	Bi-PSSM	92.70	94.59	90.63	94.76	0.8550
	CTD	94.01	95.60	92.28	95.75	0.8808
	All-XGBoost	**94.64**	96.33	92.81	96.46	0.8934

As we can see from Table 1 that for *H. pylori*, the prediction accuracy values using PAAC, AD, AAC-PSSM, Bi-PSSM, CTD, and All-XGBoost are 87.89%, 86.73%, 84.94%, 79.15%, 84.12%, and 89.27%, respectively. The ACC value of All-XGBoost is 1.38%–10.12% higher than the other five encoding schemes. All-XGBoost is at least 2.72% higher than others on MCC. For *S. cerevisiae*, the ACC values of PAAC, AD, AAC-PSSM, Bi-PSSM, CTD and All-XGBoost are 94.43%, 94.07%, 93.71%, 92.70%, 94.01% and 94.64%, respectively. After the fusion of five feature encoding methods and dimensional reduction using XGBoost achieves the highest performance. The ACC of All-XGBoost is 0.21%–1.94% higher than PAAC, AD, AAC-PSSM, Bi-PSSM and CTD. The MCC of All-XGBoost is 0.42%–3.84% higher than other single encoding methods (0.8934 vs. 0.8892, 0.8824, 0.8745, 0.8550, 0.8808). Therefore, we integrate PAAC, AD, AAC-PSSM, Bi-PSSM and CTD to obtain physicochemical property-based information, composition-based information, evolution-based information and sequence-based information.

3.3. Performance of different dimensionality reduction methods

To evaluate the effectiveness of the XGBoost feature selection method (XGBoost), we use kernel principal component analysis (KPCA) [59], singular value decomposition (SVD) [60], multidimensional scaling analysis (MDS) [61], locally linear embedding (LLE) [62], random forest (RF) [50], LightGBM [63] to reduce the dimension of the

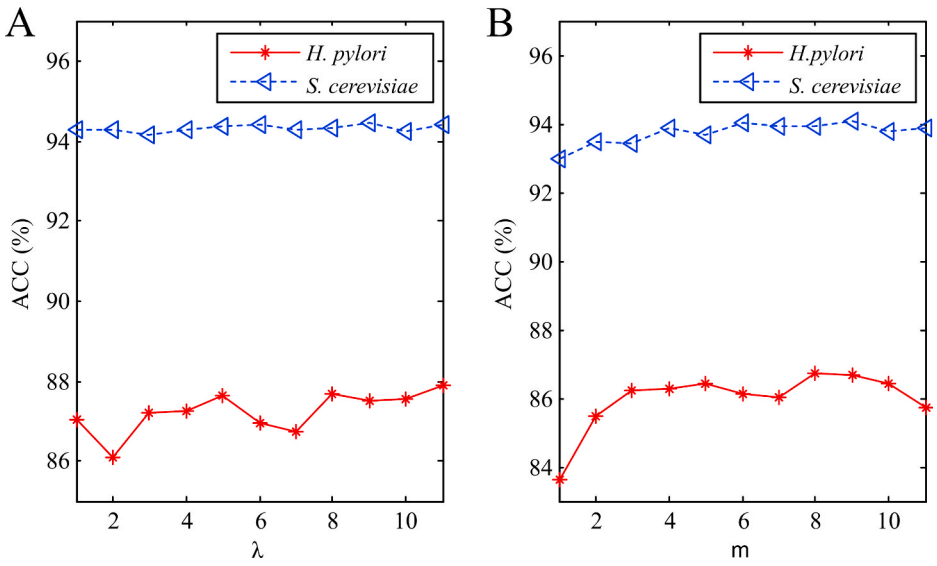

Fig. 2. Effect of selecting different values of λ and m in the feature coding process on H. pylori and S. *cerevisiae*. (A) The parameter optimization of λ in pseudo amino acid composition. (B) The parameter optimization of m in autocorrelation descriptor.

fused data. Among them, the optimal feature subset of XGBoost is 300, the contribution rate in KPCA is set as 90%, which used radial basis kernel. In this section, RF and LightGBM are employed to select the optimal feature subset. For RF, we use Gini index to split the attribute node and calculate the feature importance. For LightGBM, the variance gain is employed to split the feature node through gradient-based one-side sampling and obtain the importance score of each feature. And the numbers of optimal feature subset of RF and LightGBM are both 300. The base-classifiers of stacked ensemble classifier are two random forests and two extremely randomized trees, and the meta-classifier is a logistic regression. The prediction results of KPCA, SVD, MDS, LLE, RF, LightGBM and XGBoost are shown in Table 2. The prediction results of different dimensional reduction methods using 30 times of five-fold cross validation are listed in Table S5. And the ROC curves and the PR curves are in Fig. 3.

As we can see from Table 2, the ACC and MCC obtained by the XGBoost, KPCA, MDS, RF, LightGBM are (89.27% and 0.7859), (83.74% and 0.6753), (72.88% and 0.4579), (87.89% and 0.7582), (87.79% and 0.7561) on *H. pylori*, (94.64% and 0.8934), (85.78% and 0.7156), (65.87% and 0.3188), (93.51% and 0.8719), (93.81% and 0.8771) on *S. cerevisiae*. Among them, for *H. pylori*, the ACC of XGBoost improves 5.53% and 7.51% when compared with KPCA and SVD. The MCC of XGBoost is increased by 11.06%, 15.02%, 32.8%, 15.24%, 2.77%, and 2.98% over KPCA, SVD, MDS, LLE, RF, and LightGBM (0.7859 vs. 0.6753, 0.6357, 0.4579, 0.6335, 0.7582, 0.7561). For *S. cerevisiae*, the six dimensional reduction methods KPCA, SVD, MDS, LLE, RF, and LightGBM are inferior to XGBoost. Among them, the ACC value of XGBoost is 8.86% higher than KPCA (94.64% vs. 85.78%), 6.72% higher than SVD (94.64% vs. 87.92%), 28.77% higher than the MDS (94.64% vs. 65.87%), 9.74% higher than the LLE (94.64% vs. 84.90%), 1.13% higher than the RF (94.64% vs. 93.51%) and 0.83% higher than LightGBM (94.64% vs. 93.81%).

It can be seen from Fig. 3 (A) that on the *H. pylori* dataset, the area under the ROC curve obtained by the XGBoost is 0.9561, which is 0.75%–14.24% higher than KPCA, SVD, MDS, LLE, RF and LightGBM methods (0.9561 vs. 0.9133, 0.9010, 0.8137, 0.8973, 0.9465, 0.9486). From Fig. 3 (B), we can demonstrate that on *S. cerevisiae*, the AUC obtained by the XGBoost is 0.45%–26.44% higher than four dimensionality reduction (0.9810 vs. 0.9242, 0.9501, 0.7166, 0.9165, 0.9632, 0.9765). From Fig. 3 (C-D), on *H. pylori* and *S. cerevisiae*, XGBoost feature selection method achieves the higher AUPR value, which is superior to the prediction performance of KPCA, SVD, MDS, LLE, RF and LightGBM. The numbers of optimal subset are 300 features (*H. pylori*) and 300 features (*S. cerevisiae*), respectively.

Table 2
Prediction results of different dimensional reduction methods on *H. pylori* and *S. cerevisiae* datasets.

Dataset	Model	ACC (%)	PRE (%)	SE (%)	SP (%)	MCC
H. pylori	KPCA	83.74	83.96	83.47	84.02	0.6753
	SVD	81.76	82.95	79.97	83.54	0.6357
	MDS	72.88	73.16	72.57	73.19	0.4579
	LLE	81.66	82.33	80.59	82.72	0.6335
	RF	87.89	88.39	87.31	88.48	0.7582
	LightGBM	87.79	87.87	87.72	87.86	0.7561
	XGBoost	89.27	90.37	87.93	90.60	0.7859
S. cerevisiae	KPCA	85.78	85.92	85.59	85.97	0.7156
	SVD	87.92	88.81	86.79	61.32	0.7587
	MDS	65.87	64.55	70.43	89.06	0.3188
	LLE	84.90	85.91	83.52	86.29	0.6984
	RF	93.51	96.35	90.45	96.57	0.8719
	LightGBM	93.81	95.92	91.51	96.10	0.8771
	XGBoost	94.64	96.33	92.81	96.46	0.8934

Note: RF, LightGBM and XGBoost are employed to select the optimal feature subset.

3.4. Selection of classification algorithms

In order to select the optimal classification algorithms, we compare stacked ensemble classifier with logistic regression (LR), K-nearest neighbor (KNN) [64], AdaBoost [65], random forest (RF), support vector machine (SVM) [66] and XGBoost [46]. Among them, the neighbors of the KNN method is set as 5, radial basis kernel is utilized in SVM, the 'n_estimators' of AdaBoost, RF and XGBoost are 500, 500, and 500 respectively. Specially, in this section, the XGBoost is employed as the classifier. The prediction results are obtained via 5-fold cross-validation, which are shown in Table 3. The results of seven classifiers using 30 times of 5-fold cross-validation are listed in Table S6. The ROC and PR curves can be seen in Fig. 4. In order to further verify StackPPI, we perform a statistical test on the different classifiers. Under the $\alpha = 0.05$ significance level, we report the P-values of LR, KNN, AdaBoost, RF, SVM, XGBoost compared to stacked ensemble classifier on ACC, MCC and AUC indicators. The statistical test results obtained via the two-tailed T-test are shown in Table 4. In addition, the confidence intervals of different classifier are shown in Table 5, where Lower represents the lower bound of the confidence interval and Upper represents the upper bound of the confidence interval.

From Table 3, we can see stacked ensemble classifier is efficient and useful with an ACC of 89.27% on *H. pylori*, and with an ACC of 94.64% on *S. cerevisiae*. The ACC of logistic regression, AdaBoost, RF, SVM and XGBoost are (78.46% vs. 87.04% vs. 85.36% vs. 84.95% vs. 87.07%) on *H. pylori*, (80.91% vs. 92.14% vs. 93.18% vs. 89.18% vs. 93.72%) on *S. cerevisiae*. The MCC values of KNN, AdaBoost, SVM, XGBoost and StackPPI are *H. pylori* (0.6393 vs. 0.7418 vs. 0.6990 vs. 0.7420 vs. 0.7859), *S. cerevisiae* (0.7634 vs. 0.8435 vs. 0.7838 vs. 0.8754 vs. 0.8934). When compared with LR, KNN, AdaBoost, random forest, SVM and XGBoost, StackPPI increases by 2.2%–10.81% on *H. pylori* and 0.92%–13.73% on *S. cerevisiae* from the indicator of ACC. The MCC of StackPPI is 4.39%–21.56% higher than LR, KNN, AdaBoost, RF, support vector machine, XGBoost (0.7859 vs. 0.5703, 0.6393, 0.7418, 0.7086, 0.6990, 0.7420) and is 1.8%–27.50% higher than the other six classifiers (0.8934 vs. 0.6184, 0.7634, 0.8435, 0.8659, 0.7838, 0.8754).

Fig. 4(A and B) represents performances of seven classifiers (stack ensemble classifier, LR, KNN, AdaBoost, RF, SVM, XGBoost). It can be seen that stacked ensemble classifier has the largest AUC values on the *H. pylori* (0.9561 vs. 0.8548, 0.9119, 0.9347, 0.9357, 0.9257, 0.9430) respectively. For *S. cerevisiae*, StackPPI achieves higher AUC than LR, KNN, AdaBoost, RF, SVM (0.9810 vs. 0.8865, 0.9440, 0.9734, 0.9769, 0.9561). The comparative results indicate that StackPPI is superior to the prediction performance of logistic regression, KNN, Adaboost, RF, support vector machine and XGBoost. It can be seen from Fig. 4 (C-D) that the AUPR obtained by the stacked classifier is also the largest, which is better than the prediction performance of logistic regression, KNN, Adaboost, RF, support vector machine and XGBoost classifiers. StackPPI is 1.07%–9% higher than the other six methods (*H. pylori*).

From Table 4, under the significance level $\alpha = 0.05$, on *H. pylori* dataset, the stacked ensemble classifier statistically significantly outperforms LR, KNN, AdaBoost, RF, SVM in terms of ACC, MCC and AUC. On *S. cerevisiae* dataset, when the significance level is 0.05, the stacked ensemble classifier statistically significantly outperforms LR, KNN, AdaBoost, RF, SVM, and XGBoost. On *S. cerevisiae* dataset, when the significance level is 0.05, the prediction performance of the stacked ensemble classifier is statistically significantly better than LR, KNN, AdaBoost, RF, and SVM. The difference between StackPPI and XGBoost is not significant and we can see RF in the above sentence as P-value (AUC) is 7.76E-02, which is not less than 0.05. However, when ACC and MCC are considered, StackPPI significantly outperforms RF as $P < 0.05$. So, we employ stacked ensemble classifier as the optimal classifier to predict PPIs.

From Table 5, StackPPI achieves the optimal prediction performance. On the ACC and MCC of the *H. pylori* and *S. cerevisiae* datasets, the upper and lower bounds of the confidence interval of the StackPPI

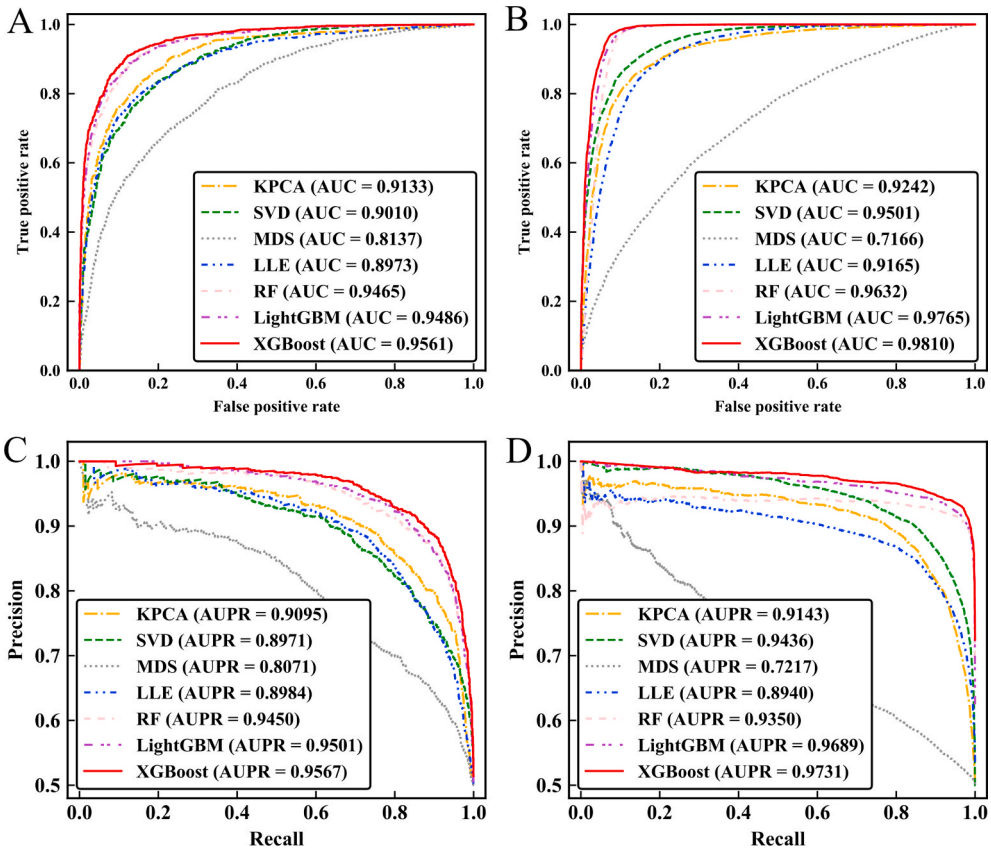

Fig. 3. The ROC and PR curves of XGBoost, KPCA, SVD, MDS, LLE dimensional reduction. (A–B) ROC curves of *H. pylori* (A) and *S. cerevisiae* (B). (C–D) PR curves of *H. pylori* (C) and *S. cerevisiae* (D).

Table 3
Prediction results of seven classifiers on *H. pylori* and *S. cerevisiae* datasets.

Dataset	Model	ACC (%)	PRE (%)	SE (%)	SP (%)	MCC
H. pylori	LR	78.46	77.15	80.94	75.99	0.5703
	KNN	81.17	75.61	92.11	70.23	0.6393
	AdaBoost	87.04	85.84	88.82	85.26	0.7418
	RF	85.36	83.74	87.86	82.85	0.7086
	SVM	84.95	84.82	85.12	84.77	0.6990
	XGBoost	87.07	86.10	88.48	85.67	0.7420
	StackPPI	89.27	90.37	87.93	90.60	0.7859
S. cerevisiae	LR	80.91	81.73	79.64	82.18	0.6184
	KNN	88.01	91.90	83.36	92.65	0.7634
	AdaBoost	92.14	93.73	90.33	93.96	0.8435
	RF	93.18	96.59	89.52	96.84	0.8659
	SVM	89.18	89.56	88.72	89.65	0.7838
	XGBoost	93.72	96.00	91.24	96.19	0.8754
	StackPPI	94.64	96.33	92.81	96.46	0.8934

Note: RF and XGBoost are employed as the classifiers.

method are higher than LR, KNN, AdaBoost, RF, SVM and XGBoost. The Lower of StackPPI is 2.37%–11.55% (*H. pylori*) and 0.45%–13.52% (*S. cerevisiae*) higher than other machine learning methods on ACC value. The Upper of StackPPI is 1.18%–10.05% (*H. pylori*) and 1.4%–13.95% (*S. cerevisiae*) higher than other machine learning methods on ACC value. From the perspective of AUC value, the upper bounds of the *H. pylori* and *S. cerevisiae* datasets are better than the other six classifiers. Considering the evaluation results, this paper selects the stacked ensemble classifier in StackPPI for PPIs prediction.

3.5. Comparing StackPPI's PPIs prediction against other state-of-the-art methods

In order to analyze the advantages and disadvantages of StackPPI, this paper lists the comparison results of StackPPI with HKNN [67], Signature products [27], Ensemble of HKNN [68], WSRC [69], PCA-EELM [70], DCT + WSRC [71], LightGBM-PPI [72] on *H. pylori* in Table 6. And the performance of StackPPI with ACC + SVM [29], LD + KNN [45], SVM + LD [30], MCD [73], WSRC [69], DeepPPI [74] and LightGBM-PPI [72] on *S. cerevisiae* are shown in Table 7.

From Table 6, StackPPI has the highest ACC of 89.27%, which is improved at least 0.24% compared with other predictors. Especially, StackPPI is 0.24%, 2.67%, 1.77% higher than LightGBM-PPI by Chen et al. [72], Ensemble of HKNN by Nanni et al. [68] and PCA-EELM by You et al. [70], respectively. The PRE and MCC values of StackPPI are 90.37% and 0.7859, respectively. StackPPI improves at least 2.01% over existing methods on PRE (90.37%). When compared with DCT + WSRC by Huang et al. [71] and LightGBM-PPI by Chen et al. [72], StackPPI is increased by 1.6% and 0.45% on MCC, respectively.

From Table 7, the accuracy, precision, sensitivity, and MCC of the StackPPI prediction model are 94.64%, 96.33%, 92.81%, and 0.8934, respectively. StackPPI's ACC value is 0.43% lower than LightGBM-PPI by Chen et al. [72]. StackPPI is 2.14% higher than WSRC by Huang et al. [69]. For SE, StackPPI is at least 0.6% higher than other PPIs predictors. The SE value of StackPPI is 0.75% higher than DeepPPI by Du et al. [74] and 2.88% higher than ACC + SVM by Guo et al. [29]. The MCC value of StackPPI is 0.37% higher than DeepPPI [74] (0.8934 vs. 0.8897), 3.25% higher than WSRC [69] (0.8934 vs. 0.8609), and 5.13% higher than MCD [73] (0.8934 vs. 0.8421).

For further verifying the StackPPI, *S. cerevisiae* (the number of interacting pairs is 11188) is employed as the training dataset to predict PPIs of *H. sapiens*, *M. musculus*, *C. elegans* and *E. coli*. First, the sequences

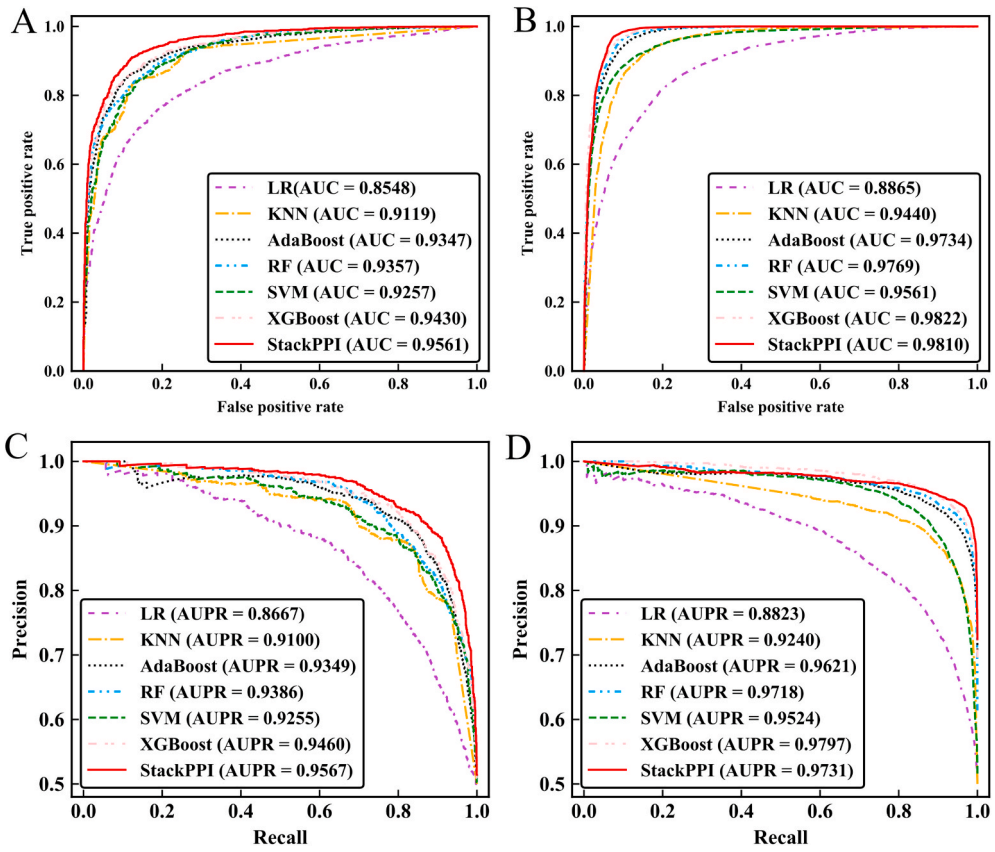

Fig. 4. The performance of LR, KNN, AdaBoost, RF, SVM, XGBoost, StackPPI using ROC and PR curves. (A–B) ROC curves of *H. pylori* (A) and *S. cerevisiae* (B). (C–D) PR curves of *H. pylori* (C) and *S. cerevisiae* (D).

Table 4
P value (T-test; two-tailed) of XGBoost with other classifiers on ACC, MCC and AUC indicators.

Dataset	Model	P-value (ACC)	P-value (MCC)	P-value (AUC)
H. pylori	LR	2.44E-06	2.59E-06	2.84E-04
	KNN	2.93E-06	3.30E-06	2.07E-04
	AdaBoost	2.34E-02	2.32E-02	8.22E-03
	RF	9.24E-04	8.75E-04	2.13E-02
	SVM	3.49E-04	3.42E-04	2.84E-05
	XGBoost	1.26E-03	1.04E-03	6.96E-03
S. cerevisiae	LR	2.69E-09	2.42E-09	1.01E-07
	KNN	1.65E-06	1.88E-06	2.70E-07
	AdaBoost	1.34E-03	1.28E-03	7.98E-03
	RF	9.09E-03	1.18E-02	7.76E-02
	SVM	1.22E-05	1.22E-05	4.29E-05
	XGBoost	5.34E-02	5.77E-02	5.36E-01

Note: RF and XGBoost are employed as the classifiers.

are encoded according to PAAC, AD, AAC-PSSM, Bi-gram PSSM, CTD, and we select 300 optimal features to train the StackPPI model. Then, the optimal feature subset information of four independent test dataset are input into the constructed StackPPI model. Finally, the prediction accuracy of four independent test sets are obtained. The comparison results of StackPPI with LightGBM-PPI [72], DPPI [75], DeepPPI [74], MLD + RF [17] (using the same dataset and experimental settings) are shown in Table 8.

From Table 8, the prediction accuracies of StackPPI on the *H. sapiens*, *M. musculus*, *C. elegans* and *E. coli* datasets are 97.66%, 98.40%, 97.11% and 98.71%, respectively, which are better than LightGBM-PPI (Chen et al.) [72], DPPI (Hashemifar et al.) [75], DeepPPI (Du et al.) [74] and MLD + RF (You et al.) [17]. There are two reasons to explain the StackPPI's excellent performance in predicting protein interactions it has not previously encountered. First, the fusion of multiple biological sources and XGBoost feature selection can effectively represent PPIs computationally. Not only that, our stacked ensemble classifier utilizes two-layered learning strategy to mine the correlation between statistical and biological information of protein pairs and labels. Secondly, predictions across species is facilitated by the presence of orthologs originating from common evolutionary ancestors. This means training a model using the interacting protein pairs from one species can be relatively easily generalized to other species possessing similar protein pairs as these will theoretically also interact.

3.6. Evaluating predictions of PPIs networks

For the binary classification task, interacting and non-interacting pairs are generated from databases accumulated via the experimental or computational approaches. So, most PPIs can be considered as graphs, where we can study the corresponding biological and topological properties. This paper uses two types of PPIs networks to perform the interactive visualization. The first network called Wnt-related pathway containing 78 genes: DVL1, FRAT1, MAPK8, AXIN1, etc. The second disease-specific network consists of 78 genes, including CDK4, MCM2, BUB1, etc. In this paper, we integrate PAAC, AD, AAC-PSSM, Bi-PSSM and CTD to encode numerical vector and utilize XGBoost to select effective and efficient features for feature representation. We train StackPPI model on *S. cerevisiae* to predict two types of networks, which can be noticed in Fig. 5 and Fig. 6, where the black solid line represents the PPIs of true prediction based on StackPPI and the red dotted line represents the false prediction.

As we can see from Fig. 5, StackPPI can successfully predict 93 of the 96 PPIs. CT + SVM by Shen et al. [76] could successfully predict 73 of the 96 PPIs, MMI + RF by Ding et al. [77] could achieve 94.79%

Table 5
The confidence intervals of evaluation metrics on *H. pylori* and *S. cerevisiae* with different classifier methods.

Dataset	Confidence interval	ACC (%)		MCC		AUC	
	Classifier	Lower	Upper	Lower	Upper	Lower	Upper
H. pylori	LR	76.94	79.99	0.5398	0.6007	0.8295	0.8794
	KNN	79.87	82.47	0.6147	0.6640	0.9005	0.9237
	AdaBoost	85.22	88.86	0.7059	0.7777	0.9222	0.9481
	RF	83.78	86.93	0.6777	0.7395	0.9209	0.9521
	SVM	83.47	86.42	0.6696	0.7285	0.9190	0.9325
	XGBoost	86.12	88.03	0.7235	0.7604	0.9352	0.9514
	StackPPI	88.49	90.04	0.7706	0.8012	0.9530	0.9602
	Classifer	Lower	Upper	Lower	Upper	Lower	Upper
S. cerevisiae	LR	80.17	81.64	0.6035	0.6334	0.8784	0.8951
	KNN	87.47	88.54	0.7527	0.7742	0.9408	0.9474
	AdaBoost	91.07	93.21	0.8223	0.8647	0.9697	0.9772
	RF	92.69	93.67	0.8557	0.8762	0.9742	0.9797
	SVM	88.76	89.61	0.7754	0.7923	0.9496	0.9625
	XGBoost	93.24	94.19	0.8660	0.8849	0.9793	0.9853
	StackPPI	93.69	95.59	0.8744	0.9123	0.9764	0.9856

Note: RF and XGBoost are employed as the classifiers.

Table 6
Comparison of StackPPI with other PPIs prediction methods on *H. pylori* dataset.

Methods	ACC (%)	PRE (%)	SE (%)	MCC
HKNN [67]	84.00	84.00	86.00	N/A
Signature products [27]	83.40	85.70	79.90	N/A
Ensemble of HKNN [68]	86.60	85.00	86.70	N/A
WSRC [69]	84.28	80.45	90.54	0.7325
PCA-EELM [70]	87.50	86.15	88.95	0.7813
DCT + WSRC [71]	86.74	87.01	86.43	0.7699
LightGBM-PPI [72]	89.03	88.36	89.99	0.7814
StackPPI	89.27 ± 0.62	90.37 ± 1.30	87.93 ± 1.78	0.7859 ± 1.23%

Note: N/A means not available, ± represents standard deviation.

Table 7
Comparison of StackPPI with other PPIs prediction methods on *S. cerevisiae* dataset.

Methods	ACC (%)	PRE (%)	SE (%)	MCC
ACC + SVM [29]	89.33 ± 2.67	88.87 ± 6.16	89.93 ± 3.68	N/A
LD + KNN [45]	86.15 ± 1.17	90.24 ± 1.34	81.03 ± 1.74	N/A
SVM + LD [30]	88.56 ± 0.33	89.50 ± 0.60	87.37 ± 0.22	0.7715 ± 0.68
MCD [73]	91.36 ± 0.36	91.94 ± 0.62	90.67 ± 0.69	0.8421 ± 0.59
WSRC [69]	92.50 ± 0.59	95.87 ± 0.89	88.82 ± 0.98	0.8609 ± 1.02
DeepPPI [74]	94.43 ± 0.30	96.65 ± 0.59	92.06 ± 0.36	0.8897 ± 0.62
LightGBM-PPI [72]	95.07 ± 0.51	97.82 ± 0.48	92.21 ± 0.61	0.9030 ± 1.01
StackPPI	94.64 ± 0.76	96.33 ± 0.80	92.81 ± 0.89	0.8934 ± 1.52

Note: N/A means not available, ± represents standard deviation.

Table 8
Comparison of StackPPI method with other state-of-the-art predictors on the independent dataset.

Species	StackPPI	LightGBM-PPI [72]	DPPI [75]	DeepPPI [74]	MLD + RF [17]
H. sapiens	97.66	94.83	96.24	93.77	94.19
M. musculus	98.40	94.57	95.84	91.37	91.69
C. elegans	97.11	90.16	95.51	94.84	87.71
E. coli	98.71	92.16	96.66	92.19	89.30

(91/96). And LightGBM-PPI by Chen et al. [72] had accuracy of 92.71% (89/96). StackPPI achieves better prediction performance than CT + SVM by Shen et al. [76], MMI + RF by Ding et al. [77] and LightGBM-PPI by Chen et al. [72]. Wnt-related pathway plays an important role in Alzheimer's disease prevention, and cancer treatment [78]. StackPPI only failed to predict three PPIs. As can be seen from Fig. 6, StackPPI can predict all 108 protein-protein interactions, successfully. CDK1 and CDK2 are valuable for cancer treatment, and CDK1 could be potential and useful selective inhibitors [79]. MCM2 can improve the transcription, which is vital in ciliogenesis [80]. Fortunately, StackPPI could successfully identify the interactions around CDK1 and MCM2.

4. Conclusion

We present a PPIs prediction algorithm, StackPPI. Biological data, including physicochemical, evolutionary, positional, and sequence derived properties obtained from public databases were encoded and fused into features via PAAC, AD, ACC-PSSM, Bi-PSSM and CTD methods. By fusing complementary features from different sources, StackPPI can garner new insights into previously undiscerned biological processes. Compared to models using single feature encoding schemes, the multi-information fusion implemented by StackPPI can more fully computationally represent PPIs, which yields improved prediction accuracy. StackPPI utilizes XGBoost for dimensionality reduction for the first time, obtaining a collection of 300 optimized features. Our results confirm XGBoost preserves key features necessary to accurately predict PPIs. StackPPI analyzes the features identified by XGBoost using its stacked ensemble classifier comprised of two RTs and two ETs base classifiers in the first stage and a LR meta-classifier in the second stage. The resultant two-layer learning model mines the feature architecture to reduce generalization errors and improve prediction accuracy. StackPPI indicates the co-evolved relationship exists in PPIs. Finally, StackPPI demonstrates immense promise when it constructed PPI networks of canonical and disease-specific pathways.

While we are able to extract various features from biological data such as physicochemical properties, evolutionary profiles and other sequence characteristics, making StackPPI a welcome improvement on the current state-of-the-art prediction models, we were unable to predict PPIs with complete accuracy. We believe the greatest strides can be made in improving our capacity to extract and utilize the structural features of proteins. Using deep learning is also a promising direction because of its capability to mine higher-level features than traditional ML models. Combining deep learning with our stacked ensemble classifier is a natural next step in the advancement of PPIs prediction.

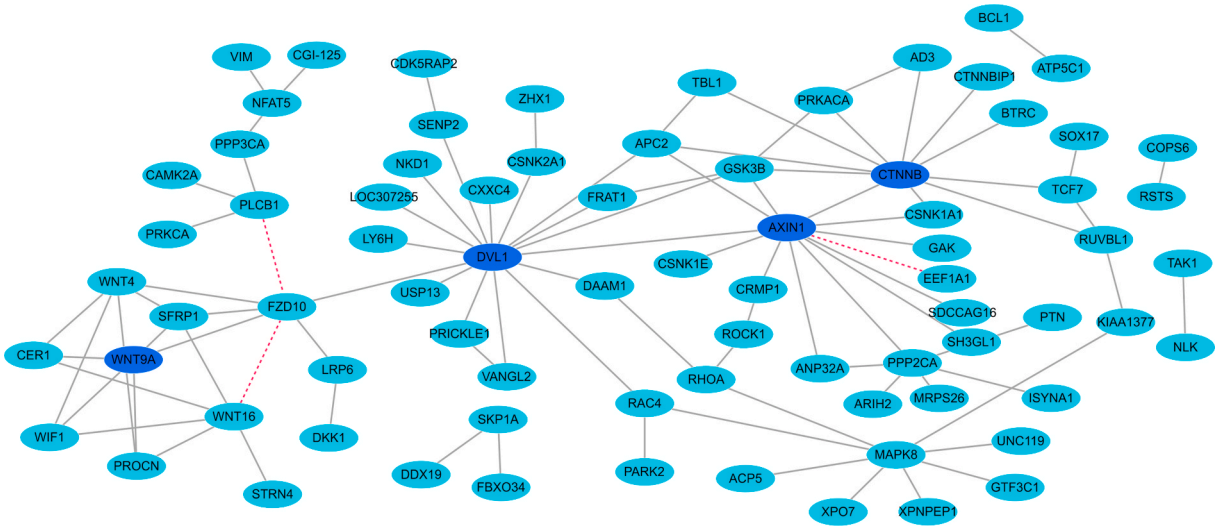

Fig. 5. The performance of StackPPI on the Wnt-related pathway crossover network. WNT9A, DVL1, AXIN1, CTNNB are linked to Wnt-related pathway.

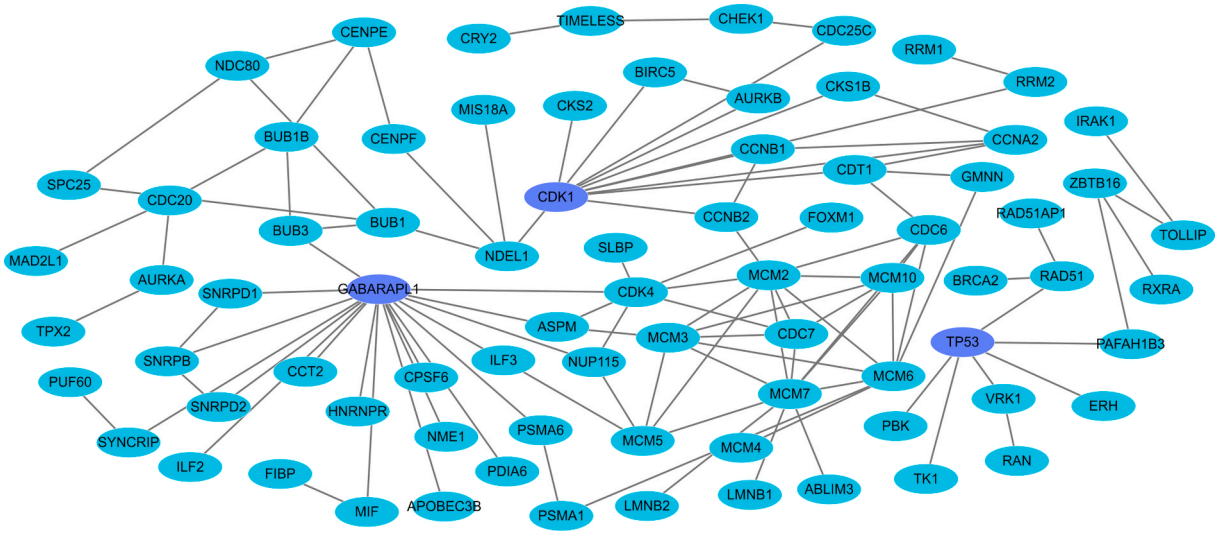

Fig. 6. The performance of StackPPI on the disease-specific network. All 108 PPIs pairs are successfully identified in network. The GABARAPL1, CDK1 and TP53 are linked to this work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank anonymous reviewers for valuable suggestions and comments. This work was supported by the National Nature Science Foundation of China (No. 61863010), the Key Research and Development Program of Shandong Province of China (No. 2019GGX101001), and the Natural Science Foundation of Shandong Province of China (No. ZR2018MC007, ZR2019MEE066).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compbiomed.2020.103899.

References

[1] D. Amar, T. Hait, S. Izraeli, R. Shamir, Integrated analysis of numerous heterogeneous gene expression profiles for detecting robust disease-specific biomarkers and proposing drug targets, Nucleic Acids Res. 43 (2015) 7779–7789.
[2] E.E. Schadt, Molecular networks as sensors and drivers of common human diseases, Nature 461 (2009) 218–223.
[3] O. Keskin, N. Tuncbag, A. Gursoy, Predicting protein-protein interactions from the molecular to the proteome level, Chem. Rev. 116 (2016) 4884–4909.
[4] H. Lin, E.Z. Deng, H. Ding, W. Chen, K.C. Chou, iPro54-PseKNC: a sequence-based predictor for identifying sigma-54 promoters in prokaryote with pseudo k-tuple nucleotide composition, Nucleic Acids Res. 42 (2014) 12961–12972.
[5] L. Wang, Z.H. You, S.X. Xia, F. Liu, X. Chen, X. Yan, Y. Zhou, Advancing the prediction accuracy of protein-protein interactions by utilizing evolutionary information from position-specific scoring matrix and ensemble classifier, J. Theor. Biol. 418 (2017) 105–110.
[6] S. Yang, H. Li, H. He, Y. Zhou, Z. Zhang, Critical assessment and performance improvement of plant-pathogen protein-protein interaction prediction methods, Briefings Bioinf. 20 (2019) 274–287.
[7] Q.C. Zhang, D. Petrey, L. Deng, L. Qiang, Y. Shi, C.A. Thu, B. Bisikirska, C. Lefebvre, D. Accili, T. Hunter, T. Maniatis, A. Califano, B. Honig, Structure-based prediction of protein-protein interactions on a genome-wide scale, Nature 490 (2012) 556–560.
[8] Z. Chen, P. Zhao, F. Li, A. Leier, T.T. Marquez-Lago, A. Leier, J. Revote, Y. Zhu, D. R. Powell, T. Akutsu, G.I. Webb, K.C. Chou, A. I Smith, R.J. Daly, J. Li, J. Song, iLearn: an integrated platform and meta-learner for feature engineering, machine-

learning analysis and modeling of DNA, RNA and protein sequence data, Briefings Bioinf. 21 (2020) 1047–1057.
[9] B. Liu, F. Liu, X. Wang, J. Chen, L. Fang, K.C. Chou, Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences, Nucleic Acids Res. 43 (2015) W65–W71.
[10] B. Yu, Y. Zhang, A simple method for predicting transmembrane proteins based on wavelet transform, Int. J. Biol. Sci. 9 (2013) 22–33.
[11] M.D. Dyer, T.M. Murali, B.W. Sobral, Computational prediction of host-pathogen protein-protein interactions, Bioinformatics 23 (2007) i159–i166.
[12] X. Lian, S. Yang, H. Li, C. Fu, Z. Zhang, Machine-learning-based predictor of human-bacteria protein-protein interactions by incorporating comprehensive host-network properties, J. Proteome Res. 18 (2019) 2195–2205.
[13] T. Sun, B. Zhou, L. Lai, J. Pei, Sequence-based prediction of protein protein interaction using a deep-learning algorithm, BMC Bioinf. 18 (2017) 277.
[14] S. Yadav, A. Ekbal, S. Saha, A. Kumar, P. Bhattacharyya, Feature assisted stacked attentive shortest dependency path based Bi-LSTM model for protein-protein interaction, Knowl.-Based Syst. 166 (2019) 18–29.
[15] R.S. Wang, Y. Wang, L.Y. Wu, X.S. Zhang, L. Chen, Analysis on multi-domain cooperation for predicting protein-protein interactions, BMC Bioinf. 8 (2007) 391.
[16] T. Hamp, B. Rost, Evolutionary profiles improve protein-protein interaction prediction from sequence, Bioinformatics 31 (2015) 1945–1950.
[17] Z.H. You, K.C.C. Chan, P.W. Hu, Predicting protein-protein interactions from primary protein sequences using a novel multi-scale local feature representation scheme and the random forest, PloS One 10 (2015), e0125811.
[18] J.Y. An, Z.H. You, F.R. Meng, S.J. Xu, Y. Wang, RVMAB: using the relevance vector machine model combined with average blocks to predict the interactions of proteins from protein sequences, Int. J. Mol. Sci. 17 (2016) 757.
[19] L. Zhang, G. Yu, M. Guo, J. Wang, Predicting protein-protein interactions using high-quality non-interacting pairs, BMC Bioinf. 19 (2018) 525.
[20] J. Wang, L. Zhang, L. Jia, Y. Ren, G. Yu, Protein-protein interactions prediction using a novel local conjoint triad descriptor of amino acid sequences, Int. J. Mol. Sci. 18 (2017) 2373.
[21] L. Zhang, G. Yu, D. Xia, J. Wang, Protein-protein interactions prediction based on ensemble deep neural networks, Neurocomputing 324 (2018) 10–19.
[22] Y.J. Ding, J.J. Tang, F. Guo, Identification of protein-protein interactions via a novel matrix- based sequence representation model with amino acid contact information, Int. J. Mol. Sci. 17 (2016) 1623.
[23] J.Y. Lin, H. Chen, S. Li, Y.S. Liu, X. Li, B. Yu, Accurate prediction of potential druggable proteins based on genetic algorithm and Bagging-SVM ensemble classifier, Artif. Intell. Med. 98 (2019) 35–47.
[24] A. Mishra, P. Pokhrel, M.T. Hoque, StackDPPred: a stacking based prediction of DNA-binding protein from sequence, Bioinformatics 35 (2019) 433–441.
[25] S. Saha, S. Mitra, R.K. Yadav, A stack-based ensemble framework for detecting cancer microrna biomarkers, Dev. Reprod. Biol. 15 (2017) 381–388.
[26] Y. Xiong, Q. Wang, J. Yang, X. Zhu, D.Q. Wei, PredT4SE-Stack: prediction of bacterial type IV secreted effectors from protein sequences using a stacked ensemble method, Front. Microbiol. 9 (2018) 2571.
[27] S. Martin, D. Roe, J.L. Faulon, Predicting protein-protein interactions using signature products, Bioinformatics 21 (2005) 218–226.
[28] I. Xenarios, L. Salwínski, X.J. Duan, P. Higney, S.M. Kim, D. Eisenberg, DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions, Nucleic Acids Res. 30 (2002) 303–305.
[29] Y. Guo, L. Yu, Z. Wen, M. Li, Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences, Nucleic Acids Res. 36 (2008) 3025–3030.
[30] Y.Z. Zhou, Y. Gao, Y.Y. Zheng, Prediction of protein-protein interactions using local description of amino acid sequence, in: Advances in Computer Science and Education Applications, 2011, pp. 254–262.
[31] U. Stelzl, U. Worm, M. Lalowski, C. Haenig, F.H. Brembeck, H. Goehler, M. Stroedicke, M. Zenkner, A. Schoenherr, S. Koeppen, J. Timm, S. Mintzlaff, C. Abraham, N. Bock, S. Kietzmann, A. Goedde, E. Toksöz, A. Droege, S. Krobitsch, B. Korn, W. Birchmeier, H. Lehrach, E.E. Wanker, A human protein-protein interaction network: a resource for annotating the proteome, Cell 122 (2005) 957–968.
[32] K.C. Chou, Prediction of protein cellular attributes using pseudo-amino acid composition, Proteins 43 (2001) 246–255.
[33] X.W. Cui, Z.M. Yu, B. Yu, M.H. Wang, B.G. Tian, Q. Ma, UbiSitePred: a novel method for improving the accuracy of ubiquitination sites prediction by using LASSO to select the optimal Chou's pseudo components, Chemomet. Intell. Lab. 184 (2019) 28–43.
[34] W.Y. Qiu, S. Li, X.W. Cui, Z.M. Yu, M.H. Wang, J. Du, Y. Peng, B. Yu, Predicting protein submitochondrial locations by incorporating the pseudo-position specific scoring matrix into the general Chou's pseudo-amino acid composition, J. Theor. Biol. 45 (2018) 86–103.
[35] B.G. Tian, X. Wu, C. Chen, W.Y. Qiu, Q. Ma, B. Yu, Predicting protein-protein interactions by fusing various Chou's pseudo components and using wavelet denoising approach, J. Theor. Biol. 462 (2019) 329–346.
[36] B. Yu, S. Li, C. Chen, J.M. Xu, W.Y. Qiu, X. Wu, R.X. Chen, Prediction subcellular localization of Gram-negative bacterial proteins by support vector machine using wavelet denoising and Chou's pseudo amino acid composition, Chemomet. Intell. Lab. 167 (2017) 102–112.
[37] B. Yu, S. Li, W.Y. Qiu, M.H. Wang, J.W. Du, Y. Zhang, X. Chen, Prediction of subcellular location of apoptosis proteins by incorporating PsePSSM and DCCA coefficient based on LFDA dimensionality reduction, BMC Genom. 19 (2018) 478.
[38] B. Yu, L.F. Lou, S. Li, Y. Zhang, W.Y. Qiu, X. Wu, M.H. Wang, B.G. Tian, Prediction of protein structural class for low-similarity sequences using Chou's pseudo amino acid composition and wavelet denoising, J. Mol. Graph. Model. 76 (2017) 260–273.
[39] X.M. Sun, T.Y. Jin, C. Chen, X.W. Cui, Q. Ma, B. Yu, RBPro-RF: use Chou's 5-steps rule to predict RNA-binding proteins via random forest with elastic net, Chemomet. Intell. Lab. 197 (2020) 103919.
[40] B. Yu, Z.M. Yu, C. Chen, A.J. Ma, B.Q. Liu, B.G. Tian, Q. Ma, DNNAce: prediction of prokaryote lysine acetylation sites through deep neural networks with multi-information fusion, Chemomet. Intell. Lab. 200 (2020) 103999.
[41] H.Y. Zhou, C. Chen, M.H. Wang, Q. Ma, B. Yu, Predicting Golgi-resident protein types using conditional covariance minimization with XGBoost based on multiple features fusion, IEEE Access 7 (2019) 144154–144164.
[42] Z. Chen, P. Zhao, F. Li, A. Leier, T.T. Marquez-Lago, Y. Wang, G.I. Webb, A.I. Smith, R.J. Daly, K.C. Chou, J. Song, iFeature: a python package and web server for features extraction and selection from protein and peptide sequences, Bioinformatics 34 (2018) 2499–2502.
[43] J. Wang, B. Yang, J. Revote, A. Leier, T.T. Marquez-Lago, G. Webb, J. Song, K. C. Chou, T. Lithgow, POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles, Bioinformatics 17 (2017) 2756–2758.
[44] S.F. Altschul, T.L. Madden, A.A. Schäffer, J. Zhang, W. Miller, D.J. Lipman, Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Res. 25 (1997) 3389–3402.
[45] L. Yang, J.F. Xia, J. Gui, Prediction of protein-protein interactions from protein sequence using local descriptors, Protein Pept. Lett. 17 (2010) 1085–1090.
[46] T. Chen, C. Guestrin, XGBoost: a scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[47] H.J. AL-barakati, H. Saigo, R.H. Newman, D.B. KC, RF-GlutarySite, Random forest based predictor for glutarylation sites, Mol. Omics. 15 (2019) 189.
[48] C. White, H.D. Ismail, H. Saigo, D.B. KC, CNN-BLPreD: a convolutional neural network based predictor for β-lactamases (BL) and their classes, BMC Bioinf. 18 (2017) 577.
[49] J. Yu, S. Shi, F. Zhang, G. Chen, M. Cao, PredGly: predicting lysine glycation sites for Homo sapiens based on XGboost feature optimization, Bioinformatics 35 (2019) 2749–2756.
[50] L. Breiman, Random forest, Mach. Learn. 45 (2001) 5–32.
[51] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Mach. Learn. 63 (2006) 3–42.
[52] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, C.J. Lin, LIBLINEAR: a library for large linear classification, J. Mach. Learn. Res. 9 (2008) 1871–1874.
[53] F. Pedregosa, G. Varo, uaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[54] K.C. Chou, Using subsite coupling to predict signal peptides, Protein Eng. 14 (2001) 75–79.
[55] B. Yu, W.Y. Qiu, C. Chen, A.J. Ma, J. Jiang, H.Y. Zhou, Q. Ma, SubMito-XGBoost: predicting protein submitochondrial localization by fusing multiple feature information and eXtreme gradient boosting, Bioinformatics 36 (2020) 1074–1081.
[56] M.H. Wang, X.W. Cui, B. Yu, C. Chen, Q. Ma, H.Y. Zhou, SulSite-GTB: identification of protein S-sulfenylation sites by fusing multiple feature information and gradient tree boosting, Neural Comput, Appl (2020), https://doi.org/10.1007/s00521-020-04792-z.
[57] H. Shi, S. Liu, J. Chen, X. Li, Q. Ma, B. Yu, Predicting drug-target interactions using Lasso with random forest based on evolutionary information and chemical structure, Genomics 111 (2019) 1839–1852.
[58] X. Wang, B. Yu, A. Ma, C. Chen, B. Liu, Q. Ma, Protein-protein interaction sites prediction by ensemble random forests with synthetic minority oversampling technique, Bioinformatics 35 (2019) 2395–2402.
[59] B. Schölkopf, A. Smola, K.R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (1998) 1299–1319.
[60] M.E. Wall, A. Rechtsteiner, L.M. Rocha, Singular value decomposition and principal component analysis, in: A Practical Approach to Microarray Data Analysis, 2002, pp. 91–109.
[61] Y.H. Taguchi, Y. Oono, Relational patterns of gene expression via non-metric multidimensional scaling analysis, Bioinformatics 21 (2005) 730–740.
[62] S.T. Roweis, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326.
[63] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.Y. Liu, LightGBM: a highly efficient gradient boosting decision tree, in: 31st Conference Neural Information Processing Systems, NeurIPS, 2017, pp. 3146–3154.
[64] F. Nigsch, A. Bender, B.V. Buuren, J. Tissen, E. Nigsch, J.B. Mitchell, Melting point prediction employing k-nearest neighbor algorithms and genetic parameter optimization, J. Chem. Inf. Model. 46 (2006) 2412–2422.
[65] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1997) 119–139.
[66] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.
[67] L. Nanni, Hyperplanes for predicting protein-protein interactions, Neurocomputing 69 (2005) 257–263.
[68] L. Nanni, A. Lumini, An ensemble of K-local hyperplanes for predicting protein-protein interactions, Bioinformatics 22 (2006) 1207–1210.
[69] Y.A. Huang, Z.H. You, X. Chen, G.Y. Yan, Improved protein-protein interactions prediction via weighted sparse representation model combining continuous wavelet descriptor and PseAA composition, BMC Syst. Biol. 10 (2016) 120.

[70] Z.H. You, Y.K. Lei, L. Zhu, J.F. Xia, B. Wang, Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis, BMC Bioinf. 14 (2013) S10.
[71] Y.A. Huang, Z.H. You, X. Gao, L. Wong, 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) 902198.
[72] C. Chen, Q.M. Zhang, Q. Ma, B. Yu, LightGBM-PPI, Predicting protein-protein interactions through LightGBM with multi-information fusion, Chemomet. Intell. Lab. 191 (2019) 54–64.
[73] Z.H. You, L. Zhu, C.H. Zheng, H.J. Yu, S.P. Deng, Z. Ji, Prediction of protein-protein interactions from amino acid sequences using a novel multi-scale continuous and discontinuous feature set, BMC Bioinf. 15 (2014) S9.
[74] X. Du, S. Sun, C. Hu, Y. Yao, Y. Yan, Y. Zhang, DeepPPI: boosting prediction of protein-protein interactions with deep neural networks, J. Chem. Inf. Model. 57 (2017) 1499–1510.
[75] S. Hashemifar, B. Neyshabur, A.A. Khan, J.B. Xu, Predicting protein-protein interactions through sequence-based deep learning, Bioinformatics 34 (2018) i802–i810.
[76] J. Shen, J. Zhang, X. Luo, W. Zhu, K. Yu, K. Chen, Y. Li, H. Jiang, Predicting protein-protein interactions based only on sequences information, Proc. Natl. Acad. Sci. U. S. A. 104 (2007) 4337–4341.
[77] Y. Ding, J. Tang, F. Guo, Predicting protein-protein interactions via multivariate mutual information of protein sequences, BMC Bioinf. 17 (2016) 398.
[78] M. Katoh, M. Katoh, Molecular genetics and targeted therapy of WNT-related human diseases (Review), Int. J. Mol. Med. 40 (2017) 587–606.
[79] N.R. Brown, S. Korolchuk, M.P. Martin, W.A. Stanley, R. Moukhametzianov, M.E. M. Noble, J.A. Endicott, CDK1 structures reveal conserved and unique features of the essential cell cycle CDK, Nat. Commun. 6 (2015) 6769.
[80] T.C. Tena, L.D. Maerz1, K. Szafranski, M. Groth, T.J. Blatte, C. Donow, S. Matysik, P. Walther, P.A. Jeggo, M.D. Burkhalter, M. Philipp, Resting cells rely on the DNA helicase component MCM2 to build cilia, Nucleic Acids Res. 47 (2019) 134–151.