Computers in Biology and Medicine 134 (2021) 104516

Contents lists available at ScienceDirect

Computers in Biology and Medicine

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

Prediction of protein-protein interaction sites through eXtreme gradient boosting with kernel principal component analysis

Xue Wang [a,b,1], Yaqun Zhang [a,b,1], Bin Yu [a,b,c,*], Adil Salhi [d], Ruixin Chen [a,b], Lin Wang [a,b], Zengfeng Liu [a,b]

[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] Key Laboratory of Computational Science and Application of Hainan Province, Haikou, 571158, China
[d] Computational Bioscience Research Center (CBRC), King Abdullah University of Science and Technology (KAUST), Thuwal, 23955, Saudi Arabia

ARTICLE INFO

Keywords:
Protein-protein interaction sites
Feature extraction
SMOTE
KPCA
XGBoost

ABSTRACT

Predicting protein-protein interaction sites (PPI sites) can provide important clues for understanding biological activity. Using machine learning to predict PPI sites can mitigate the cost of running expensive and time-consuming biological experiments. Here we propose PPISP-XGBoost, a novel PPI sites prediction method based on eXtreme gradient boosting (XGBoost). First, the characteristic information of protein is extracted through the pseudo-position specific scoring matrix (PsePSSM), pseudo-amino acid composition (PseAAC), hydropathy index and solvent accessible surface area (ASA) under the sliding window. Next, these raw features are preprocessed to obtain more optimal representations in order to achieve better prediction. In particular, the synthetic minority oversampling technique (SMOTE) is used to circumvent class imbalance, and the kernel principal component analysis (KPCA) is applied to remove redundant characteristics. Finally, these optimal features are fed to the XGBoost classifier to identify PPI sites. Using PPISP-XGBoost, the prediction accuracy on the training dataset Dset186 reaches 85.4%, and the accuracy on the independent validation datasets Dtestset72, PDBtestset164, Dset_448 and Dset_355 reaches 85.3%, 83.9%, 85.8% and 85.4%, respectively, which all show an increase in accuracy against existing PPI sites prediction methods. These results demonstrate that the PPISP-XGBoost method can further enhance the prediction of PPI sites.

1. Introduction

Protein-protein interaction (PPI) enables proteins to play their biological function and provides the basis for biochemical reactions within organisms. DNA replication and transcription, gene expression regulation, cell metabolism, signal transduction, enzyme catalysis, confirmation of immune response, as well as protein synthesis, are all the results of various protein-protein interaction mechanisms [1]. Understanding protein-protein interaction sites (PPI sites) is necessary for the process of identifying them and can provide important clues for understanding PPI function. This is in turn helpful for studies on drug design [2], signal transduction networks [3], protein interaction network reconstruction, etc. However, with advances in the human genome project, there is an unprecedented increase in the amount of data being deposited to protein sequence databases. Relying exclusively on experimental approaches such as double hybridization of yeast [4], mass spectrometry [5], and other traditional biological mass spectrometry analysis methods to predict PPI sites, is becoming ever more impractical. Computational approaches, and in particular machine learning, which incorporates a wealth of techniques that lie in the border between statistical analysis and computer science, can guide biological experiments [6]. Machine learning also holds the promise of elucidating an understanding of the biological implications contained in data. The application of machine learning in bioinformatics research has been shown to effectively improve the prediction accuracy across a number of applications including the study of PPI sites, and has recently gained a wider practical value [7].

Extracting features that are effective for classification from raw representations, is the key to successfully applying machine learning for PPI sites prediction. In recent years, researchers carried out a series of

* 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.2021.104516
Received 29 December 2020; Received in revised form 24 May 2021; Accepted 24 May 2021
Available online 1 June 2021

studies on extracting and encoding features from protein sequences. Such efforts typically fuse physicochemical properties, sequence information, evolutionary information, structural information, functional annotation, as well as others, in order to obtain a rich and comprehensive set of protein features [8]. Murakami et al. [9] extracted characteristic information based on position-specific scoring matrix (PSSM) and predicted accessibility (PA). Dhole et al. [10] conducted feature extraction by integrating PSSM, averaged cumulative hydropathy (ACH) and predicted relative solvent accessibility (PRSA). Zhang et al. [11] applied various feature extraction methods including 3D-1D scores, conservation score, and protein sequence coding. Chou [12] used the pseudo-amino acid composition (PseAAC) method to encode protein sequences and classify the interaction data of three types of proteins. Dong et al. [13] combined the sequence spectrum and solvent accessible surface area to identify residues on the interaction interface and achieved good results. Xie et al. [14] fused amino acid encoding, PSSM, position specific frequency matrix (PSFM), composition-transition-distribution (CTD), and five structure features to encode protein sequences. Chen et al. [15] combined PseAAC, autocorrelation descriptor (AD), conjoint triad (CT), and local descriptor (LD) features, to predict PPI. Zeng et al. [16] used the n-gram frequency method to obtain three features from protein sequences. The features were then combined in a weighting scheme to make full use of the sequence information. Göktepe et al. [17] fused weighted skip-sequential conjoint triads, PSSM, Bi-Gram representation, AAC and PseAAC to extract information features from protein sequences, and achieved good results. GcForest-PPI [18] adopted PseAAC, autocorrelation descriptor (Auto), multivariate mutual information (MMI), CTD, amino acid composition PSSM (AAC-PSSM) and dipeptide composition PSSM (DPC-PSSM) to extract features for prediction. Furthermore, a variety of improved methods based on PseAAC have been applied to extract protein sequence information. For example, Nanni et al. [19] used a sequence-based algorithm combining the augmented Chou's pseudo amino acid composition based on the autocovariance approach. Göktepe et al. [20] proposed a method based on the weighted amino acid composition information and the correlation factors of the protein. Xu et al. [21] incorporated two tiers of sequence pair coupling effects into the general form of PseAAC to improve the predictive ability of the model.

Biological information hidden in protein sequences needs to be identified and predicted by high-throughput, high-accuracy prediction algorithms. Therefore, the selection of classifiers is crucial in the prediction of PPI sites. Common classification algorithms include logistic regression (LR) [22], support vector machine (SVM) [23], neural network [24], Naïve Bayes (NB) [25], conditional random field (CRF) [26], decision tree (DT) [27] and random forest (RF) [28]. Ofran and Rost [29] applied a neural networks approach to predict PPI sites from sequence information. Neuvirth et al. [30] used primo Bayes to identify potential action sites of protein structures in the monomer state. Yan et al. [31] were the first to apply the SVM algorithm to the identification of PPI sites, and later developed a two-stage classifier combining SVM and NB to further improve the prediction. Murakami et al. [9] used kernel density estimation to train a Bayes classifier to improve prediction performance. Dhole et al. [10] predicted PPI sites through combining a neural network and regularized LR for training. Wei et al. [32] and Wang et al. [33] identified PPI sites through the RF classifier. Wei et al. [34] applied integrated SVM and weighted RF for the prediction of PPI sites. Zeng et al. [35] used a text convolutional neural network to extract features from protein sequence and applied the deep neural network to predict PPI sites.

Although the existing machine learning based methods have achieved good results in the prediction of PPI sites, there are still many problems to be addressed. First, the extreme class imbalance between positive and negative samples may negatively affect prediction results and hinder the predictive ability of traditional machine learning algorithms [36–38]. Second, on the one hand, a single feature extraction method cannot comprehensively represent the sequence information. On the other hand, fusing multiple features in order to enrich sequence information, although beneficial, does introduce redundancy and noise to the features. Finally, we find that existing classifiers for PPI sites prediction, such as RF, NB, and LR, can still be improved upon.

To address the problems mentioned above, PPISP-XGBoost is proposed in this paper as a new and effective method for PPI sites prediction. First, in order to incorporate characteristic information from sequence, structure, as well as the physicochemical properties of the protein, we fused the following features: pseudo-position specific scoring matrix (PsePSSM), pseudo amino acid composition (PseAAC), hydropathy index, and solvent accessible surface area (ASA). Next, we used the synthetic minority oversampling technique (SMOTE) to minimize the effect of class imbalance, and subsequently applied kernel principal component analysis (KPCA) to remove the effect of noise and redundancy, and only retain the effective characteristics. In terms of the prediction of PPI sites, this paper also introduces the use of XGBoost as the classifier for the first time.

The results of 10-fold cross-validation show that the prediction performance of XGBoost on the extracted features is superior to the NB, SVM, Adaptive boosting (AdaBoost), DT, multilayer perceptron (MLP), gradient boosting decision tree (GBDT), K nearest neighbor (KNN) and RF classifiers on the training dataset Dset186. Furthermore, four independent validation datasets, Dtestset72, PDBtestset164, Dset_448 and Dset_355, also achieved satisfactory prediction results under PPISP-XGBoost, where the ACC reached 85.3%, 83.9%, 85.8% and 85.4%, the MCC reached 0.712, 0.685, 0.718 and 0.857, the F-measure reached 0.851, 0.839, 0.709 and 0.853, respectively. Experimental results indicated that the PPISP-XGBoost method can enhance the prediction accuracy of PPI sites.

2. Materials and methods

2.1. Datasets

To construct a PPI sites prediction model and compare it with other existing prediction methods, five datasets are used in this paper. The Dset186 [9] is used for training the model, and the other four: Dtestset72 [9], PDBtestset164 [38], Dset_448 [39] and Dset_355 [40] are used as independent validation datasets to evaluate the performance of the proposed approach. Dset186 and Dtestset72 are both extracted from the protein data bank (PDB) database [41] by Murakami et al. Dset186 contains 186 non-homologous and non-redundant protein sequences with less than 25% sequence homology, and the structure of these proteins has a resolution of less than 3.0 A under X-ray crystallography. Dtestset72 contains 72 non-redundant protein sequences which are composed of 36 protein complexes and have no overlap with Dset186. PDBtestset164 contains 164 non-redundant protein sequences obtained from the new annotated proteins in the PDB database with the same filtering process as Dset186 and Dtestset72. Dset_448, which is built from the BioLip [42] database by Zhang et al. contains 448 protein sequences. Dset_448 was made to have less than 25% sequence similarity by removing protein fragments, mapping BioLip sequences to UniProt sequences, and clustering. Finally, Dset_355 was obtained by deleting 93 sequences from Dset_448 to mitigate the impact of redundancy [40].

For Dset186, Dtestset72, and PDBtestset164, residues with less than 5% of relative accessible surface area (RASA) [43] are defined as surface residues, and the lost surface residues with absolute solvent accessibility greater than 1.0 $Å^2$ are defined as interacting residues during the formation of protein complexes [44]. Based on the above definition, protein structure and interaction analyzer (PSAIA) software [45] identified the interacting residues of the three datasets. There are 5517, 1923, 6096 interacting residues and 30702, 16217, 27585 non-interacting residues on Dset186, Dtestset72 and PDBtestset164. If the distance between one atom of a residue and another is less than 0.5 Åplus the sum of the Van der Waals radii, the two residues are defined as interacting

residues. There are 15810, 11467 interacting residues and 100690, 84473 non-interacting residues on Dset_448 and Dset_355. As a result, the proportions of interacting residues are 15.2%, 10.6%, 18.1%, 13.6% and 12.0% in the five datasets, respectively.

2.2. Feature extraction

In this paper, 120-dimensional pseudo-position specific score matrix (PsePSSM) features were extracted based on protein evolution information, additionally, 26-dimensional pseudo-amino acid composition (PseAAC) features were extracted based on protein sequence information. Considering the physicochemical properties and structural characteristics of proteins, 9-dimensional hydropathy index and 9-dimensional solvent accessible surface area (ASA) features were extracted for the prediction of PPI sites. The details are shown in Supplementary Table S1. Since most of the residues located at the protein-protein interaction interface may be clustered in their adjacent local sequences, the extraction of local sequence signals plays a significant role in the identification of PPI sites [30]. Therefore, we applied the sliding window technique and combined the above four methods to capture the feature information of target residues and their adjacent residues. After many tests, the size of the sliding window was set to 9, and the detailed comparative results are shown in Supplementary Table S2.

2.2.1. Pseudo-position specific scoring matrix

Position-specific scoring matrix (PSSM) [46] is a very common and effective feature extraction method, but it is only suitable for the case of equal sequence length in the dataset to be processed. Due to the operational problems caused by the varying length of sequences, a new feature representation method named PsePSSM [47] was proposed. At present, the PsePSSM method has been extensively utilized in many aspects of bioinformatics, such as identification of multi-label protein subcellular localization [48] and prediction of subcellular location of apoptosis proteins [49]. PsePSSM is produced by translating PSSM, which not only reflects sequence evolution information but also effectively deals with the unequal sequence length in the dataset to be processed. PSI-BLAST [50] was applied to search the database. For PSI-BLAST, we used the three iterations and set *E*-value to 0.001 [51–53].

The constructed PSSM format is shown in formula (1):

$$P_{PSSM} = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,20} \\ \vdots & \vdots & & \vdots \\ p_{i,1} & p_{i,2} & \cdots & p_{i,20} \\ \vdots & \vdots & & \vdots \\ p_{L,1} & p_{L,2} & \cdots & p_{L,20} \end{pmatrix} \tag{1}$$

where $p_{i,j}$ expresses the logarithmic likelihood score of the i-th amino acid in row i and column$j$ mutating to j-th class. L represents the amount of amino acids in the protein. The rows of the PSSM matrix show the corresponding amino acid positions in the sequence, and the columns show the 20 amino acid types that may be mutated.

First, formula (2) maps the original data of the PSSM matrix to the range [0,1] to make the influence weights of each dimension on the objective function consistent, which could effectively improve the convergence speed of iteration and solution. Details are shown in Supplementary Table S3. Where x is the original data of the PSSM matrix.

$$f(x) = 1/(1+e^{-x}) \tag{2}$$

Second, the vector dimension of the PSSM matrix is unified by formula (3):

$$\overline{P_{PSSM}} = \left(\overline{P_1}, \overline{P_2}, \cdots, \overline{P_{20}}\right) \tag{3}$$

where the elements $\overline{P_j}$ are obtained by formula (4):

$$\overline{P_j} = \sum_{i=1}^{L} P_{i,j} \,/\, L, (j = 1, 2, \cdots, 20) \tag{4}$$

$\overline{P_j}$ represents the average score of amino acid residues which are mutated to j amino acid type in the protein.

A protein sequence can then be described by PsePSSM as follows:

$$P_{PsePSSM} = \left(\overline{P_1}, \overline{P_2}, \cdots, \overline{P_{20}}, \theta_1^1, \theta_2^1, \cdots, \theta_{20}^1, \cdots, \theta_1^\xi, \theta_2^\xi, \cdots, \theta_{20}^\xi\right)^T \tag{5}$$

The elements θ_j^ξ are obtained by formula (6):

$$\begin{aligned} &\theta_j^{\,\xi} = \tfrac{1}{L-\xi} \sum_{i=1}^{L-\xi} \left(P_{i,j} - P_{(i+\xi),\,j}\right)^2, \\ &(j = 1,\ 2,\ \cdots,\ 20,\ \xi < L,\ \xi \neq 0) \end{aligned} \tag{6}$$

where θ_j^ξ represents the sequence information of the protein sequence, j represents the amino acid type, and ξ is the adjacent distance of the amino acid. On this basis, the PsePSSM algorithm is used to generate $20 + 20 \times \xi$-dimensional feature vectors.

2.2.2. Pseudo-amino acid composition

The traditional amino acid composition (AAC) reflects the composition information of amino acids by calculating the frequencies of 20 amino acids in the sequence. Chou [12] put forward the PseAAC method, which incorporates both sequence and position information between the residues, in order to produce better features for the protein sequence. At present, the PseAAC algorithm has been extensively utilized in many aspects of bioinformatics, such as the identification of prokaryote lysine acetylation sites [54], prediction of protein submitochondrial locations [55], prediction of protein-protein interactions [56], etc.

A protein X with a sequence length of L can be expressed as a feature vector with $20 + \lambda$ dimensions by using the PseAAC algorithm:

$$X = [x_1, x_2, \ldots, x_{19}, x_{20}, x_{20+1}, \ldots, x_{20+\lambda}]^T, (\lambda < L) \tag{7}$$

As shown in formula (7), the first 20-dimensional feature vectors are composed of traditional AAC, and the other λ values express the sequence information of amino acid residues.Here,

$$x^u = \begin{cases} \dfrac{f_u}{\sum_{i=1}^{20} f_i + \omega \sum_{j=1}^{\lambda} \theta_j}, & (1 \le u \le 20) \\ \dfrac{\omega \theta_{u-20}}{\sum_{i=1}^{20} f_i + \omega \sum_{j=1}^{\lambda} \theta_j}, & (20+1 \le u \le 20+\lambda) \end{cases} \tag{8}$$

where θ_j describes the sequence correlation factor of the j-th layers and represents the sequence information of two adjacent amino acids. f_i represents the appearance frequency of the 20 amino acids and ω represents the weight coefficient.

2.2.3. Hydropathy index

"Hydropathy index" was proposed by Kyte and Doolittle in 1982 [57]. Hydrophobicity is a common physical property, which is composed of the transfer free energy of each amino acid measured in the experiment. It can describe the degree of hydrophilicity or hydrophobicity of a certain amino acid branch chain, which is crucial for understanding the energy of protein two-layer interactions [58]. The hydrophobic value is a scalar. The larger the value is, the more hydrophobic the residue is, conversely, the smaller the value is, the more hydrophilic the residue is. The specific values used are adopted from a paper by Wimley and White [59]. The two amino acids with the strongest hydrophobicity are isoleucine (hydropathy index 4.50) and valine (hydropathy index 4.20). The two most hydrophilic amino acids are

Table 1
Hydropathy indexes of 20 common amino acids.

Amino acids	A	C	D	E	F	G	H	I	K	L
Values	1.80	2.50	−3.50	−3.50	2.80	−0.40	−3.20	4.50	−3.90	3.80
Amino acids	M	N	P	Q	R	S	T	V	W	Y
Values	1.90	−3.50	−1.60	−3.50	−4.50	−0.80	−0.70	4.20	−0.90	−1.30

arginine (hydropathy index −4.50) and lysine (hydropathy index −3.90). In order to extract protein sequence information more conveniently, Table 1 lists the hydropathy indexes of the 20 common amino acids.

2.2.4. Accessible surface area

The surface shape of protein molecules is so irregular that the analysis of its surface properties is of great significance in the study of PPI sites and molecular docking. Lee and Richards [60] firstly proposed the concept of ASA, which quantitatively describes the burying of hydrophobic amino acids of proteins by calculating the surface area of the proteins contacted by water molecules. ASA, as an important one-dimensional structure feature, not only highlights the role of internal residues on the surface of the protein but also reflects the spatial structure distribution of the protein sequence to a certain extent [61].

2.3. Synthetic minority oversampling technique

Generally speaking, unbalanced samples could prompt the training model to focus on the categories with a large number of samples, thus affecting the generalization capability of the model on the test data. Since the difference between positive and negative samples in our datasets is large, the accuracy of predicting protein-protein interaction sites may be affected. In order to reduce the prediction error, the synthetic minority oversampling technique (SMOTE) algorithm, which is used to deal with imbalanced samples, is used in this paper. The SMOTE algorithm is an intelligent oversampling technique put forward by Chawla et al. [62], which can mitigate the overfitting of the classifier caused by the shrinkage introduced by traditional oversampling. SMOTE plays an important role in many applications, such as the prediction of drug-target interaction [63], the prediction of the s-sulfonyl sites [64], and the prediction of RNA binding proteins [65]. The fundamental idea of SMOTE is to balance the class of the sample by using linear interpolation to increase the number of small samples. Setting the oversampling multiple of the sample as N, first, samples are randomly selected from the K homogeneous neighbors of each minority sample. And then according to formula (9), each minority sample and its corresponding selected samples are combined into new minority samples. Finally adding them to the original training dataset to form a new training sample set:

$$X_{new} = X + rand[0,1] \times (y[i] - x), (i = 1, 2, \cdots, n) \tag{9}$$

where $rand[0,1]$ represents a random real between 0 and 1, x represents minority samples. X_{new} represents the new sample of synthesis and $y[i]$ is the i-th nearest neighbor of x. After processing the five datasets with SMOTE, sample imbalance is addressed, and we proceed to the next step.

2.4. Kernel principal component analysis

Kernel principal component analysis (KPCA) [66] is a nonlinear extension of the principal components analysis (PCA) [67] method. This method performs nonlinear dimension reduction based on the kernel technique, and has good results in the extraction of nonlinear statistics and higher-order statistical features. PCA is usually unable to reduce the dimensionality of data in a contextual sense and cannot process nonlinear data. The KPCA algorithm, however, can mine the nonlinear information contained in the dataset. Besides allowing nonlinear dimension reduction of the data, KPCA can also process datasets that are linearly inseparable in the original input space.

Assuming that vectors $\omega_i\ (i = 1, \cdots, d)$ are the eigenvectors of the high-dimensional representation, with the corresponding eigenvalues $\lambda_i\ (i = 1, \cdots, d)$. PCA in the high-dimensional space is as follows:

$$\Phi(X)\Phi(X)^T\omega_i = \lambda_i \tag{10}$$

(i) The eigenvector $\omega_i(i = 1, \cdots, d)$ is linearly represented by the sample set $\Phi(X)$, and then is substituted into $\omega_i(i = 1, \cdots, d)$, then by multiplying both sides of the equation by $\Phi(X)^T$, we get the following formula:

$$\Phi(X)^T\Phi(X)\Phi(X)^T\Phi(X)\alpha = \lambda_i\ \Phi(X)^T\Phi(X)\alpha \tag{11}$$

(ii) The purpose of this is to construct two $\Phi(X)^T\Phi(X)$, which are further replaced by the kernel matrix K (symmetric matrix). It is also an important step in the KPCA method. Thus, the formula is further changed into the following form:

$$K^2\alpha = \lambda_i K\alpha \tag{12}$$

(iii) Where α is the eigenvector of K. By removing K from both sides, an exceeding high PCA similarity solution formula is obtained:

$$K\alpha = \lambda_i\alpha \tag{13}$$

Considering formula (11), the eigenvector ω in the high-dimensional space should be further calculated from α. Since the basis $\omega_i(i = 1, \cdots, d)$ of a higher-dimensional space can be obtained, this basis can constitute a subspace of the higher-dimensional space. The purpose, namely the core of the KPCA method, is to obtain the linear representation of the test sample x_{new} in this subspace. In other words, the purpose is to obtain the vector $\widehat{x}_{new}$ after dimension reduction. The process of dimension reduction is as follows:

$$\begin{aligned}\widehat{x}_{new} &= \omega_i^T x_{new} = \left[\sum_{i=1}^{N}\Phi(x_i)\alpha_i\right]^T\Phi(x_{new}) \\ &= (\alpha_1, \cdots, \alpha_N)[k(x_1, x_{new}), \cdots, k(x_N, x_{new})]^T\end{aligned} \tag{14}$$

2.5. eXtreme gradient boosting

eXtreme gradient boosting (XGBoost) [68] is an ensemble learning method based on GBDT [69] that can be used for both classification and regression problems. The method is widely used in many fields, including predicting golgi-resident protein types [70], predicting protein submitochondrial localization [71], and predicting N6-methylation sites in mRNAs [72]. Unlike the GBDT algorithm, which only used the first-order derivative information, the XGBoost algorithm also added the second-order derivative to customize the loss function. By adding regularizing terms to simplify the learning model and control the complexity of the prediction, over-fitting is effectively prevented. The algorithm supports parallelization of split points, which also greatly improves the training speed.

For a given dataset $D = \{(x_i, y_i)\}, (|D| = n, x_i \in \mathrm{R}^m, y_i \in \mathrm{R}^n)$ with n samples and m features, the prediction process is described as follows:

$$\widehat{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{15}$$

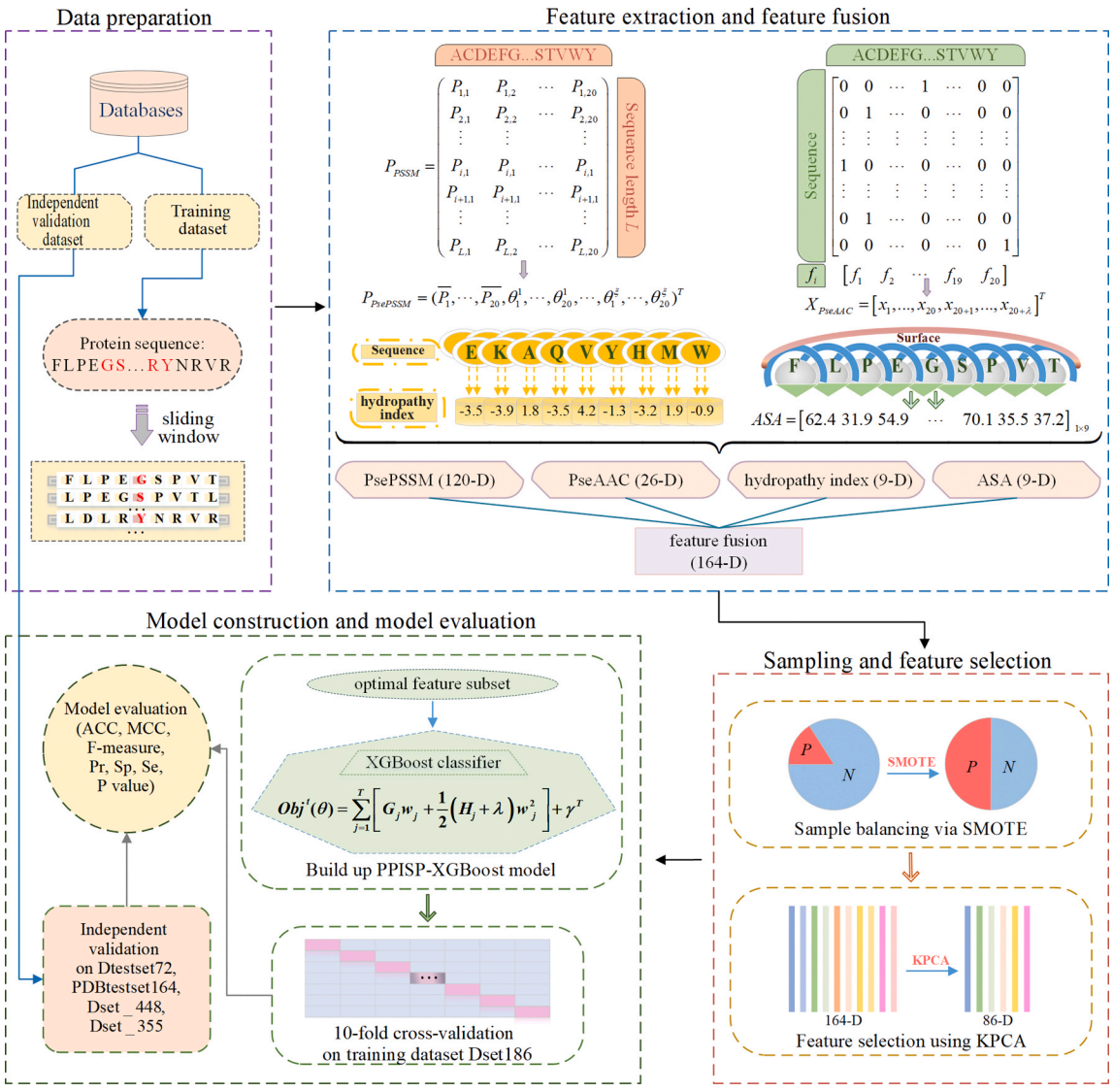

Fig. 1. The prediction framework of PPISP-XGBoost. The specific steps of the PPISP-XGBoost method for identifying PPI sites are described as follows:
Step 1: *Data preparation.* One training dataset and four independent validation datasets were gathered for model construction and performance evaluation, respectively. The sliding window technique was used for the preliminary processing of the protein sequences.
Step 2: *Feature extraction and feature fusion.* Four methods were used for feature extraction, namely, PsePSSM, PseAAC, hydropathy index, and ASA, and the initial feature subset was obtained by fusing the resulting multi-dimensional feature vectors.
Step 3: *Sampling and feature selection.* The SMOTE algorithm was used to obtain a balanced initial feature subset. The KPCA algorithm was applied to eliminate redundant features. Then the optimal feature subsets were prepared for the prediction of PPI sites.
Step 4: *Model construction and model evaluation.* The generated feature vectors were fed into the XGBoost classifier to construct the prediction model. Based on 10-fold cross-validation, the Dset186 was used to train the model, and evaluate the model performance on the training data. Finally, four independent validation datasets Dtestset72, PDBtestset164, Dset_448 and Dset_355 were also used to verify the effectiveness of the PPISP-XGBoost model.

where k represents the total number of trees and $f_k(x_i)$ represents the prediction score of the sample x_i on the k-th tree. XGBoost is an ensemble learning method which uses the space of regression trees (CART) as its base classifier functions, so the prediction score of the XGBoost algorithm can also be expressed by the above formula, and the objective function can be defined as follows:

$$Obj^t(\theta) = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i\right) + \sum_{k=1}^{K} \Omega(f_k) \tag{16}$$

where $l(y_i, \hat{y}_i)$ represents the training error of the sample x_i. In these boosting methods, the k-th tree is added to complete the t-th iteration and the prediction function is defined as:

$$\hat{y}_i^t = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{t-1} + f_k(x_i) \tag{17}$$

where $\hat{y}_i^t$ represents the prediction result of the combined t tree models on the sample x_i, the $l(y_i, \hat{y}_i^{t-1})$ of the t-th tree is a constant, and $\Omega(f_k)$ is used to describe the complexity of the k-th tree as the regularizing term, expressed as follows:

$$\Omega(f_k) = \gamma^T + \frac{1}{2}\lambda \sum_{j=1}^{t} w_j^2 \tag{18}$$

where γ and λ are the regularization parameters, and w is the score of the leaf nodes. Then the model can be written as $f_t(x) = w_{q(x)}, w \in R^t$ for

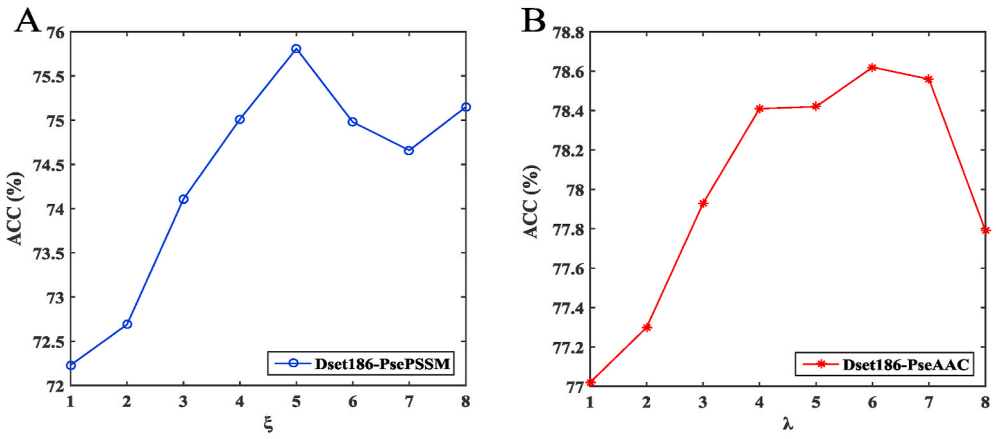

Fig. 2. The results of different ξ (A) and λ (B) on Dset186.

each regression tree. $q(x)$ indicates the leaf nodes corresponding to the sample x, and T is the number of leaf nodes of the tree.

The first derivative g_i and the second derivative h_i are simultaneously used to approximate the function using Taylor's expansion. Then the objective function can be converted into the form of the leaf node of the t-th tree by combining the above formulas and using the equality $f_t(x) = w_{q(x)}, w \in R^t$. The solution process is described as follows:

$$\begin{aligned} Obj^t(\theta) &= \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{t-1} + f_k(x_i)\right) + \Omega(f_k) + C \\ &\simeq \sum_{j=1}^{T}\left[G_j w_j + \frac{1}{2}(H_j + \lambda) w^2{}_j\right] + \gamma^T \end{aligned} \tag{19}$$

where,

$$\begin{aligned} G_j &= \sum_{i \in I_j} g_i = \sum_{i \in I_j} \partial_{\hat{y}_i^{t-1}} l\left(y_i, \hat{y}_i^{t-1}\right), \\ H_j &= \sum_{i \in I_j} h_i = \sum_{i \in I_j} \partial^2_{\hat{y}_i^{t-1}} l\left(y_i, \hat{y}_i^{t-1}\right) \end{aligned} \tag{20}$$

Then, the optimal weights w can be reflected in the first step g and the second step h, and obtained as follows:

$$w_j^* = G_i/(H_i + \lambda) \tag{21}$$

2.6. Performance assessment and model building

At present, cross-validation is one of the most widely applied techniques for model evaluation. With k-fold cross-validation, all data samples are randomly split into k disjoint sub-collections with similar sizes. One of these subsets is used for testing, and the others are set aside for training. This is repeated for all possible selections in the process until all the subsets are tested. Subsequently, the final prediction result is obtained by averaging the test results across the individual splits. After many experiments, 10-fold cross-validation is chosen to evaluate the prediction performance of the PPISP-XGBoost model proposed in this study.

Accuracy (ACC), precision (Pr), sensitivity (Se), specificity (Sp), Matthew's correlation coefficient (MCC), and F-measure were used to evaluate the above methods. The definitions of various indicators are defined as follows [33]:

$$ACC = \frac{TN + TP}{TN + TP + FN + FP} \tag{22}$$

$$Pr = \frac{TP}{TP + FP} \tag{23}$$

$$Se = \frac{TP}{TP + FN} \tag{24}$$

$$Sp = \frac{TN}{TN + FP} \tag{25}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN) \times (TN + FP) \times (TP + FP) \times (TN + FN)}} \tag{26}$$

$$F - measure = 2 \times \frac{Se \times Pr}{Se + Pr} \tag{27}$$

where *TP* is the number of sites which are correctly predicted as interaction sites, *FN* is the number of sites which are incorrectly predicted as non-interacting sites, *FP* stands for the amount of sites which are incorrectly predicted as interacting sites, and *TN* stands for the number of sites that were correctly predicted as non-interacting sites. In addition, the area covered under the ROC curve (AUC value) and the area covered under the PR curve (AUPR value) are also added as evaluation indicators to verify the robustness of the model.

For the convenience of presentation, the details of the proposed algorithm are shown in Fig. 1, where the PPI sites prediction method proposed by this study is referred to as PPISP-XGBoost.

The experimental environment for the presented results was Intel (R) Xeon (TM) i5-3210 M CPU @ 2.50 GHz with 32.0 GB of memory, MATLAB 2015a, Python 3.6, and the R Studio programming implementation. The source codes and datasets are available at https://github.com/QUST-AIBBDRC/PPISP-XGBoost/.

3. Results and discussion

3.1. Selection of optimal parameters ξ and λ

The built-in parameters of the feature extraction methods affect the reflected feature information and the dimension of the feature vector directly. Therefore, the tuning of these parameters is very important, as a wrong choice of some of these parameters may considerably affect the prediction of PPI sites. For avoiding prediction errors as much as possible and mining effective feature information, parameters were tuned for two of the feature extraction methods. The parameter values of PsePSSM and PseAAC feature extraction algorithms are represented by ξ and λ as described above. In order to select the optimal values for these parameters, the training dataset Dset186 was also used for parameter search. These two parameters were tuned independently by varying each within the range of integer values from 1 to 8. The 8 corresponding feature vectors for each parameter is obtained by extracting feature information on sequences. After using the SMOTE method for imbalance processing, these feature vectors are fed into the XGBoost for classifi-

Table 2
Comparison of prediction results with different feature extraction algorithms on Dset186.

Algorithms	ACC (%)	Pr (%)	Se (%)	Sp (%)	MCC	F-measure
PsePSSM	75.81	74.26	78.35	73.28	0.5223	0.7595
PseAAC	78.54	83.88	69.81	88.28	0.5991	0.7251
hydropathy index	74.28	79.77	59.01	89.55	0.5169	0.6128
ASA	73.73	70.62	80.81	66.65	0.4826	0.7526
Fusion	**80.99**	87.35	72.00	89.98	**0.6498**	**0.7617**

Note: The best results are given in bold.

cation. Finally, six evaluation indicators measured under 10-fold cross-validation are applied as test standards to select the optimal parameters. The prediction results of PPI sites obtained for different choices of ξ and λ are shown in Supplementary Tables S4 and S5. The corresponding ACC results of the training dataset Dset186 under different values of ξ and λ are shown in Fig. 2.

Fig. 2 depicts how changes in the feature selection parameters affect the prediction results of the corresponding models. As we can see in Fig. 2 (A) for PsePSSM, there is an initial upward trend for $\xi < 5$, the performance then gradually deteriorates for $\xi < 5$. The model at $\xi = 5$, reaches peak performance, where Dset186 obtains the largest ACC and F-measure. In Fig. 2 (B) for PseAAC, the ACC reaches its highest point at $\lambda = 6$ after which the model performance starts to fall for Dset186. Based on the above, the parameter values $\xi = 5$ and $\lambda = 6$ were chosen as optimal, and used to construct the model. Therefore, when employing the PsePSSM and the PseAAC method to capture sequence feature information, the dimensions of the feature vectors are $20 + 20 \cdot \xi = 120$ and $20 + \lambda = 26$, respectively.

3.2. Influence of the feature extraction algorithm on results

The eigenvectors acquired through a single feature extraction method have potential deviation, and therefore, cannot adequately represent the feature information of the PPI sites. Considering the conservative nature of the protein sequence, as well as sequence and structure characteristics, we chose four methods, PsePSSM, PseAAC, hydropathy index and ASA, for feature extraction from Dset186. The integration of these extracted features by fusing the four characteristics of the initial features provides more comprehensive information. In order to investigate the influence of the different feature extraction methods and feature fusion on PPI sites prediction, both single features and the final feature space Fusion are fed into XGBoost to predict PPI sites on Dset186, and 10-fold cross-validation is used to evaluate the model. The prediction results for Dset186 based on different single features, and feature set Fusion, are shown in Table 2. In addition, this paper also compares the prediction results for Dtestset72 and PDBtestset164 based on different single features and their fusion, and the details are shown in Supplementary Tables S6–S7.

As we can see from Table 2, the accuracy of prediction varies according to the feature extraction method used. For Dset186, PsePSSM, PseAAC, hydropathy index, ASA and Fusion obtain a prediction accuracy of 75.81%, 78.54%, 74.28%, 73.73% and 80.99%, respectively. The ACC for PseAAC is 78.54%, which is 2.73%, 4.26% and 4.81% higher than PsePSSM, hydropathy index and ASA, respectively. PseAAC has the best prediction performance compared with the other three methods. The ACC for Fusion is 80.99%, which is 2.45% higher than that of PseAAC, 5.18% higher than that of PsePSSM, 6.71% higher than that of hydropathy index and 7.26% higher than that of ASA. The MCC for Fusion is 12.75%, 5.07%, 13.29% and 16.72% higher than PsePSSM, PseAAC, hydropathy index and ASA, respectively. The comparison results indicate that using feature fusion yielded superior prediction results to those with single features, therefore, feature fusion can improve the prediction of PPI sites to some extent. Hence, PsePSSM, PseAAC, hydropathy index and ASA are fused within our model to represent the features extracted from the various datasets in this paper.

Table 3
Comparison of prediction results with different treatments of sample imbalance on Dset186.

Algorithms	ACC (%)	Pr (%)	Se (%)	Sp (%)	MCC	F-measure
unbalance	52.51	51.36	54.64	50.38	0.0511	0.5253
TL	53.27	52.73	54.81	51.74	0.6421	0.6592
RUS	56.71	55.79	65.31	48.10	0.1359	0.6015
NM	62.19	61.22	66.90	57.49	0.2464	0.6376
SMOTE	**80.99**	87.35	72.00	89.98	**0.6498**	**0.7617**

Note: The best results are given in bold.

3.3. Comparison of unbalanced and balanced samples

The extreme imbalance in the number of positive and negative samples may lead to the prediction results being skewed in favor of majority samples, which greatly reduces the performance of traditional machine learning algorithms and affects the identification accuracy of PPI sites. Therefore, four kinds of unbalanced data treatment algorithms are applied to balance the samples, in particular, Tomek links (TL) [73], random under sampler (RUS) [74], NearMiss (NM) [75], and SMOTE. Consequently, four balanced eigenvectors are obtained after processing. In order to analyze the impact of using unbalanced versus balanced samples on PPI sites prediction, the original unbalanced set of feature vectors in Dset186 and the four balanced sets obtained using the four algorithms mentioned above, are fed into the classifier for prediction comparison. The results are shown in Table 3.

It can be seen from Table 3 that, for Dset186, after using the four algorithms, various evaluation indexes are significantly improved. The indexes obtained for the SMOTE algorithm reach the optimal effect, and the value of ACC, MCC and F-measure reach 80.99%, 0.6498 and 0.7617, respectively. Its prediction accuracy is 28.48%, 27.72%, 24.28% and 18.80% higher than that of unprocessed unbalanced samples, respectively. The MCC and F-measure are 0.77%–51.39% and 10.25%–16.02% higher than the other three sampling methods, respectively. It can be seen that the SMOTE algorithm has the best effect.

From the prediction results of the three datasets, it can be seen that SMOTE greatly improves the accuracy of prediction. Therefore, we chose SMOTE to deal with the imbalance in the characteristic vectors and used the resulting balanced set of characteristic vectors as input to the next step.

3.4. Selection of dimension reduction methods

Different dimension reduction methods have different effects on PPI sites prediction. To choose the best dimension reduction method and determine the best feature subset, this paper selects a total of 8 dimension reduction methods for comparison, namely KPCA [66], independent validation component analysis (ICA) [76], logistic regression (LR) [77], minimum absolute contraction and selection operator (LASSO) [78], spectral embedding (SE) [79], factor analysis (FA) [80], mutual information (MI) [81] and principal component analysis (PCA) [82]. A 164-dimensional feature vector is obtained after fusing four feature extraction methods on Dset186. After treating class imbalance, the aforementioned eight-dimension reduction methods are applied in turn. Eight feature vectors with different dimensionalities are obtained, which are put into the XGBoost classifier, and the prediction accuracy is obtained using 10-fold cross-validation. Table 4 shows the eigenvector dimensions obtained by the different dimension reduction methods and their influence on the prediction work. In addition, Fig. 3 shows the ROC and PR curves for the eight dimension reduction methods.

It can be seen from Table 4 that the prediction accuracy for Dset186 is different when different dimension reduction methods are used. For

Table 4
Comparison of prediction results under different dimension reduction methods on Dset186.

Algorithms	Dim	ACC (%)	Pr (%)	Se (%)	Sp (%)	MCC	F-measure
Raw	164	80.99	87.35	72.00	89.98	0.6498	0.7617
PCA	10	74.00	73.33	72.71	75.29	0.4823	0.7414
MI	40	77.53	83.04	87.36	67.70	0.5822	0.7097
SE	60	78.46	80.12	74.84	82.09	0.5807	0.7643
FA	15	78.94	78.46	78.39	79.49	0.5823	0.7856
LASSO	45	79.48	84.34	87.64	71.31	0.6161	0.7467
LR	160	80.75	87.03	89.81	71.70	0.6449	0.7583
ICA	50	82.05	81.20	80.94	83.15	0.6482	0.8176
KPCA	86	**85.37**	84.58	86.54	84.20	**0.7077**	**0.8554**

Note: The best results are given in bold.

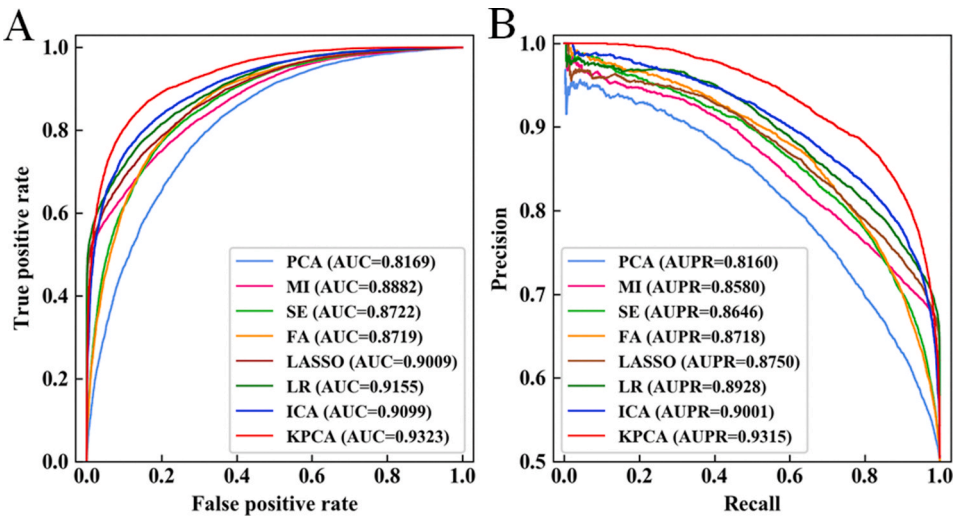

Fig. 3. The ROC and PR curves for different dimension reduction methods on Dset186.

KPCA, the values of ACC, MCC and F-measure are 85.37%, 0.7077 and 0.8554. Where ACC is 11.37%, 7.84%, 6.91%, 6.43%, 5.89%, 4.62% and 3.32% higher than the other seven methods.

As we can see in Fig. 3 (A), for Dset186, the ROC curve coverage area corresponding to KPCA is significantly larger than that of the other 7 feature selection methods. At this time, the predicted AUC value is 0.9323, which is 1.68%–11.54% higher than that of the other 7 methods. It can be seen from the PR curve in Fig. 3 (B) that the curve with the largest coverage area corresponds to the dimension reduction method of KPCA, and the curves corresponding to the other 7 dimension reduction methods are all located below the curve of KPCA. In terms of the indicator AUPR, the corresponding AUPR value of KPCA reaches 0.9315, which is 3.14%–11.55% higher than that of the other 7 methods.

For Dset186, KPCA was the most effective dimension reduction technique according to the comprehensive analysis of ROC and PR curves, and the various evaluation indexes. Therefore, the final dimension reduction method that was selected for this paper is KPCA.

3.5. The selection of classification algorithms

The choice of the classification algorithm plays an indispensable role in model construction. Here, we compare XGBoost with Naïve Bayes (NB) [25], support vector machines (SVM) [83], AdaBoost [84], decision tree (DT) [85], multilayer perceptron (MLP) [86], GBDT [69], K nearest neighbor (KNN) [87] and random forest (RF) [88] on Dset186, in order to select the best classification algorithms for predicting PPI sites. The parameters of NB are kept at default values. The kernel function is implanted in the SVM algorithm. The default penalty term L2 is used for the regularization of the LR algorithm. The learning rate of the AdaBoost algorithm is fixed to 0.1 and the number of iterations is set to 1000. The parameters of the DT are kept at default values. The number of nodes in the hidden layers of the MLP algorithm are specified at 25 and 1. The learning rate of the GBDT algorithm is set to 0.01, and the number of iterations is set to 500. The number of neighbors in the KNN algorithm is set to 5. The number of decision trees in the RF algorithm is set to 400. The learning rate of the XGBoost algorithm is set to 0.01, and the number of iterations to 500. The optimal feature subcollection picked by KPCA is fed into the above 9 classifiers, and the 10-fold cross-validation method is applied to obtain the prediction results for Dset186 using the different classifiers as illustrated in Table 5. The ROC curve and the PR curve are shown in Fig. 4. In addition, we also compare the prediction results of these classification algorithms on Dtestset72 and PDBtestset164. The results are illustrated in Supplementary Tables S8–S9, and the ROC curve and PR curve are included in Supplementary Figs. S1–S2.

As shown in Table 5, the XGBoost algorithm achieves the best

Table 5
Comparison prediction results of the classification algorithms on Dset186.

Classifiers	ACC (%)	Pr (%)	Se (%)	Sp (%)	MCC	F-measure
NB	61.95	60.82	62.91	61.00	0.2429	0.6143
SVM	62.00	62.10	60.34	63.61	0.2404	0.6108
AdaBoost	64.63	63.99	64.84	64.43	0.2959	0.6406
DT	70.83	70.54	71.29	70.36	0.4182	0.7079
MLP	70.66	67.40	79.40	61.92	0.4329	0.7237
GBDT	70.48	69.89	70.94	70.03	0.4142	0.7005
KNN	73.80	66.75	94.79	52.81	0.5252	0.7833
RF	79.81	77.85	82.86	76.75	0.6049	0.7990
XGBoost	**85.37**	84.58	86.54	84.20	**0.7077**	**0.8554**

Note: The best results are given in bold.

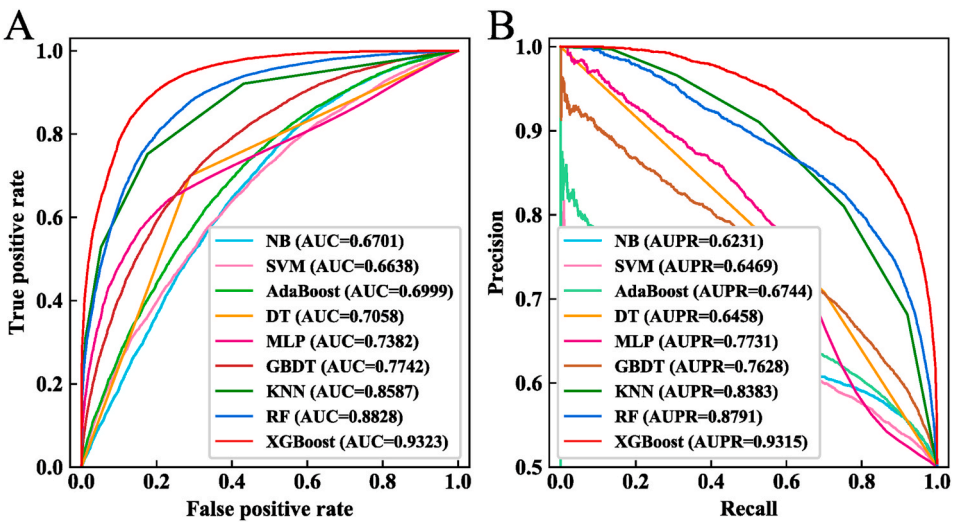

Fig. 4. The ROC and PR curves for different classification algorithms on Dset186.

Table 6
P values of RF and other classifiers on ACC, MCC, AUC (two-tailed T-test).

Dataset	Methods	P-value (ACC)	P-value (MCC)	P-value (AUC)
Dset186	NB	5.26E-06	6.67E-06	4.46E-05
	SVM	1.44E-08	1.42E-08	9.60E-08
	AdaBoost	2.26E-06	2.69E-06	1.37E-05
	DT	4.80E-08	5.88E-08	2.16E-09
	MLP	6.74E-06	5.43E-05	6.85E-08
	GBDT	2.47E-05	3.13E-05	9.63E-05
	KNN	1.14E-09	2.94E-07	1.14E-04
	RF	2.35E-02	4.09E-02	4.76E-02

performance on Dset186 with the highest values of ACC, MCC and F-measure, which are 85.37%, 0.7077 and 0.8554, respectively. Compared with other classification algorithms, the ACC, MCC and F-measure of XGBoost achieve corresponding improvements of 5.56%–23.42%, 10.28%–46.73% and 5.64%–24.46%. The above results show that the XGBoost algorithm has the best prediction effect on Dset186.

As shown in Fig. 4 (A), Dset186 has the largest coverage area under the ROC curve corresponding to XGBoost, and its AUC achieves the maximum value of 0.9323, which is 4.95%–26.85% better than NB, SVM, AdaBoost, DT, MLP, GBDT, KNN and RF. It is also evident from the PR curve in Fig. 4 (B) that Dset186 has the largest coverage area under the PR curve corresponding to XGBoost as well, and the AUPR achieves the maximum value of 0.9315, which is 5.24%–30.84% higher than other classifiers. Furthermore, in order to demonstrate the impact of k-fold cross-validation with different k values on the selection of the classification algorithm, we compare the prediction results for k in 2, 4, 6, 8 and 10. Regardless of which cross-validation method was used, XGBoost shows the best predictive performance compared to other classifiers. The ACC, MCC, F-measure of XGBoost are higher than those of NB, SVM, AdaBoost, DT, MLP, GBDT, KNN, and RF. The prediction results of the nine classification algorithms in the light of different cross-validation on Dset186 are included in Supplementary Tables S10–S14 and Supplementary Fig. S3.

In addition to the above evaluation, we also introduce the evaluation index of a two-tailed T-test [89] and add confidence intervals for further evaluating the advantages and disadvantages of PPISP-XGBoost under the $\alpha = 0.05$ significance level. The performance of the classification algorithm is further tested by analyzing the difference between its results obtained by the other classifiers. For this purpose, we use two indicators. The P values representing the significance in difference with the classifiers NB, SVM, AdaBoost, DT, MLP, GBDT, KNN, and RF on the indicators ACC, MCC, and AUC are shown in Table 6. And the confidence

Table 7
The confidence intervals of evaluation metrics on Dset186 with different classifier methods.

Confidence interval	ACC (%)		MCC		AUC (%)	
Classifiers	Lower	Upper	Lower	Upper	Lower	Upper
NB	56.32	67.59	0.1279	0.3579	58.89	75.12
SVM	58.91	65.05	0.1793	0.3016	62.34	70.42
Adaboost	60.07	69.19	0.2036	0.3882	63.78	76.20
DT	68.26	72.91	0.3666	0.4604	68.26	72.91
MLP	66.94	74.37	0.3444	0.5214	70.97	76.67
GBDT	66.13	74.84	0.3259	0.5025	72.01	82.84
KNN	72.31	75.28	0.4890	0.5613	83.26	88.47
RF	75.20	84.42	0.5077	0.7021	83.39	93.16
PPISP-XGBoost	**84.72**	**86.02**	**0.6948**	**0.7206**	**92.87**	**93.59**

Note: The best results are given in bold.

intervals of the different classifiers are shown in Table 7, where Lower and Upper correspond to the lower and upper bounds of the confidence interval.

It can be seen from Table 6 that after using the two-tailed T-test to process Dset186, XGBoost is statistically significantly different from other classifiers at the significance level of $\alpha = 0.05$. The P values of the three indicators, ACC, MCC and AUC, are all less than 0.05, which indicates that the performance of XGBoost is significantly better than NB, SVM, AdaBoost, DT, MLP, GBDT, KNN and RF.

Table 7 shows that PPISP-XGBoost achieves the best prediction performance. For the ACC, MCC and AUC on Dset186, the upper and lower bounds of the confidence intervals of the PPISP-XGBoost are higher than NB, SVM, Adaboost, DT, MLP, GBDT, KNN and RF. The lower bounds of PPISP-XGBoost are 9.52%–28.40% and 18.71%–56.69% higher than other classifiers on ACC and MCC values. The upper bounds of PPISP-XGBoost are 1.60%–20.97% and 1.85%–41.90% higher than other methods on ACC and MCC values. From the perspective of AUC values, the upper bound on Dset186 is better than these other classifiers. Considering these evaluation results, we selected XGBoost as the classifier of choice for PPI sites prediction within PPISP-XGBoost.

By comprehensively comparing the prediction results on Dset186 under nine different classification algorithms, XGBoost achieved better prediction than the NB, SVM, AdaBoost, DT, MLP, GBDT, KNN and RF classifiers. Furthermore, it can be said that XGBoost shows higher robustness and stronger generalization ability. Considering the faster training speed, the higher accuracy, and the advantage of avoiding overfitting, the XGBoost algorithm is chosen as the best classifier to build the model.

Table 8
Performance comparison of PPISP-XGBoost and other prediction methods on Dset186.

Methods	ACC (%)	Pr (%)	Se (%)	Sp (%)	MCC	F-measure
PSIVER [9]	67.3	30.6	41.6	74.3	0.151	0.353
LORIS [10]	60.4	28.7	69.8	58.6	0.221	0.384
CRF [32]	66.2	31.8	61.2	67.4	0.235	0.390
SSWRF [34]	67.9	32.2	58.1	69.7	0.234	0.386
DLPred [11]	71.8	28.5	55.6	74.7	0.238	0.377
Xie's method [14]	67.0	66.9	67.3	66.7	0.340	0.671
EL-SMURF [33]	79.1	77.7	81.7	76.6	0.584	0.784
PPISP-XGBoost	**85.4**	84.6	86.5	84.2	**0.708**	**0.855**

Note: The best results are given in bold.

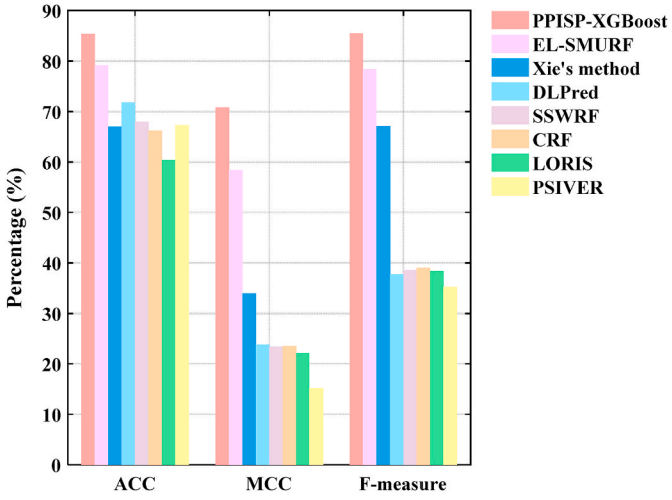

Fig. 5. ACC, MCC and F-measure of different prediction methods on Dset186.

3.6. Comparisons with other methods

At present, a variety of prediction methods have been applied to the prediction of PPI sites. In order to objectively assess the actual prediction capability of the PPISP-XGBoost model proposed in this study, we compare it with PSIVER [9], LORIS [10], DLPred [11], SPPIDER [28], CRF [32], EL-SMURF [33], SSWRF [34], SPRINGS [38], Xie's method [14] and DeepPPISP [35].

3.6.1. Comparisons with other methods on training dataset Dset186

As shown in Table 8, the ACC of the PPISP-XGBoost model acting on Dset186 is 85.4%, which is 6.3%–25.0% higher than the other seven methods, and improves 17.5% on SSWRF, 13.6% on DLPred, 18.4% on Xie's method, and 6.3% on EL-SMURF. At the same time, the MCC, F-measure, Pr, Se and Sp all reach the maximum value in comparison. Where the MCC reaches 0.708, which is 12.4%–55.7% higher than other methods, and the F-measure reaches 0.855, which is 7.1%–50.2% higher than the others.

Fig. 5 shows the values of ACC, MCC and F-measure on Dset186 under different PPI sites prediction methods. It can be seen from Fig. 5 that when PPISP-XGBoost is applied on Dset186, the ACC, MCC and F-measure are significantly higher than the other seven methods, namely, PSIVER, LORIS, CRF, SSWRF, DLPred, Xie's method and EL-SMURF. In particular, MCC and F-measure are significantly improved. The above analysis shows that satisfactory prediction results are obtained on Dset186 using the PPISP-XGBoost model proposed in this study.

Table 9
Performance comparison of PPISP-XGBoost and other prediction methods on Dtestset72.

Methods	ACC (%)	Pr (%)	Se (%)	Sp (%)	MCC	F-measure
SPPIDER [28]	61.7	20.4	45.4	63.7	0.081	0.241
PSIVER [9]	66.1	25.0	46.5	69.3	0.135	0.278
SPRINGS [38]	62.4	24.1	59.0	63.0	0.170	0.318
LORIS [10]	61.4	23.8	63.1	61.0	0.117	0.324
CRF [32]	64.0	25.6	64.0	64.0	0.209	0.340
SSWRF [34]	64.8	26.7	65.4	64.3	0.351	0.224
DeepPPISP [35]	65.5	29.8	57.7	No	0.201	0.397
DLPred [11]	73.1	40.1	55.3	68.9	0.298	0.465
Xie's method [14]	66.1	66.6	64.4	67.8	0.322	0.655
EL-SMURF [33]	77.1	76.2	78.8	75.3	0.542	0.775
PPISP-XGBoost	**85.3**	84.6	86.2	84.5	**0.712**	**0.851**

Note: The best results are given in bold.

Table 10
Performance comparison of PPISP-XGBoost and other prediction methods on PDBtestset164.

Methods	ACC (%)	Pr (%)	Se (%)	Sp (%)	MCC	F-measure
SPPIDER [28]	71.6	23.1	16.2	85.1	0.015	0.129
PSIVER [9]	59.6	25.3	46.4	63.4	0.078	0.295
SPRINGS [38]	60.6	26.8	40.7	64.8	0.108	0.311
LORIS [10]	58.8	26.3	53.8	60.9	0.111	0.323
CRF [32]	61.3	32.3	54.3	64.5	0.155	0.370
SSWRF [34]	62.1	32.3	52.7	65.6	0.152	0.365
DLPred [11]	71.1	31.2	49.1	76.0	0.214	0.381
Xie's method [14]	64.4	64.1	65.6	63.2	0.288	0.648
EL-SMURF [33]	77.7	76.4	75.3	80.0	0.554	0.782
PPISP-XGBoost	**83.9**	83.1	83.1	84.8	**0.685**	**0.839**

Note: The best results are given in bold.

Table 11
Performance comparison of PPISP-XGBoost and other prediction methods on Dset_448.

Methods	ACC (%)	Pr (%)	Se (%)	Sp (%)	MCC	F-measure
SPPIDER [28]	78.1	19.4	20.2	87.0	0.071	0.198
PSIVER [9]	78.3	19.1	19.1	87.4	0.066	0.191
SPRINGS [38]	79.6	22.8	22.9	88.2	0.111	0.229
LORIS [10]	80.5	26.3	26.4	88.7	0.151	0.263
CRF [32]	80.5	26.4	26.8	88.7	0.154	0.266
SSWRF [34]	81.1	28.6	28.8	89.1	0.178	0.287
DELPHI [40]	82.9	37.1	37.1	90.1	0.272	0.371
PPISP-XGBoost	**85.8**	86.0	85.6	86.1	**0.718**	**0.857**

Note: The best results are given in bold.

Table 12
Performance comparison of PPISP-XGBoost and other prediction methods on Dset_355.

Methods	ACC (%)	Pr (%)	Se (%)	Sp (%)	MCC	F-measure
SPPIDER [28]	80.4	18.0	18.0	88.9	0.068	0.180
PSIVER [9]	80.3	17.7	17.8	88.8	0.065	0.177
SPRINGS [38]	81.1	21.0	21.1	89.2	0.103	0.211
LORIS [10]	81.8	24.0	24.2	89.6	0.137	0.241
CRF [32]	81.9	24.5	24.7	89.7	0.143	0.246
SSWRF [34]	82.5	26.8	26.8	90.1	0.168	0.268
DLPred [11]	83.5	30.8	30.8	90.6	0.214	0.308
DELPHI [40]	84.8	36.4	36.4	91.4	0.278	0.364
PPISP-XGBoost	**85.4**	85.0	85.9	84.9	**0.709**	**0.853**

Note: The best results are given in bold.

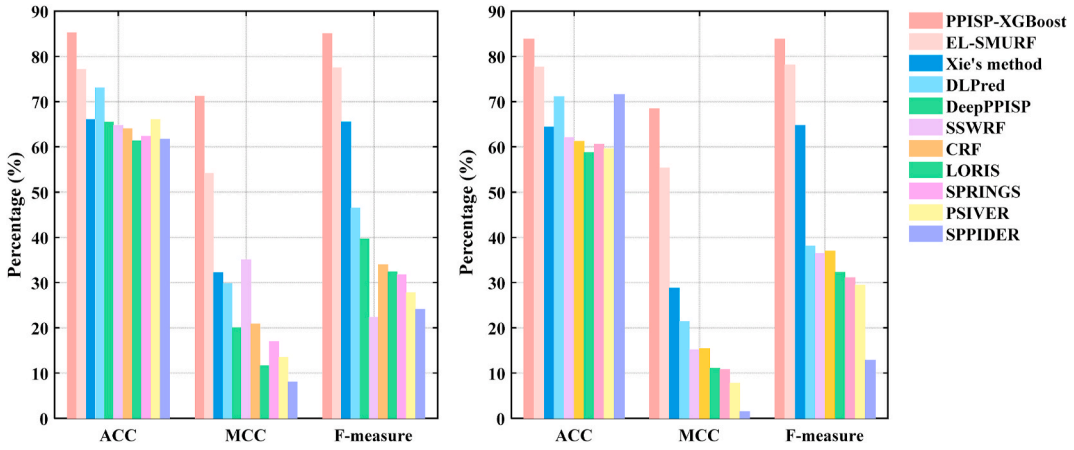

Fig. 6. ACC, MCC and F-measure of different prediction methods on Dtestset72 and PDBtestset164.

3.6.2. Comparisons with other methods on four independent validation datasets Dtestset72, PDBtestset164, Dset_448 and Dset_355

Independent validation datasets Dtestset72, PDBtestset164, Dset_448 and Dset_355 are used for testing the PPISP-XGBoost model for interaction sites prediction, in order to further evaluate the generalizability of the developed model to unseen data. The prediction results of four independent validation datasets under different PPI sites prediction methods are shown in Tables 9–12.

As shown in Table 9, the ACC of the PPISP-XGBoost on Dtestset72 reaches 85.3%, which is 8.2%–23.9% higher than other methods. Compared with LORIS and EL-SMURF, the ACC is, respectively improved by 23.9% and 8.2%. The MCC and F-measure of the PPISP-XGBoost are also significantly improved, reaching 0.712 and 0.851, which corresponds to a 17.0%–63.1% and 7.6%–62.7% increase compared to other methods.

As shown in Table 10, by comparing the prediction results to these nine methods on the independent validation dataset PDBtestset164, the prediction performance of the PPISP-XGBoost model is shown to be superior again. The ACC of the PPISP-XGBoost reaches 83.9% on PDBtestset164 which is a 6.2%–25.1% improvement over other methods. The MCC and F-measure of PPISP-XGBoost reach 0.685, and 0.839, which corresponds to 13.1% and 5.7% improvement over EL-SMURF.

Fig. 6 shows the values for ACC, MCC and F-measure on Dtestset72 and PDBtestset164 under nine different PPI sites prediction methods. It can be seen from Fig. 6 that the ACC, MCC and F-measure on Dtestset72 and PDBtestset164 all reach the highest values under the PPISP-XGBoost. Compared with SPPIDER, PSIVER, SPRINGS, LORIS, CRF, SSWRF, DLPred, Xie's method and EL-SMURF, three indicators are significantly improved for XGBoost. It can be said that the PPISP-XGBoost model can achieve satisfactory prediction results on the two independent validation datasets, Dtestset72 and PDBtestset164.

Table 11 shows the prediction results of PPISP-XGBoost alongside seven other methods for comparison on dataset Dset_448. The ACC value of PPISP-XGBoost reaches 85.8%, which is 2.9%–7.7% higher than other methods. The MCC and F-measure of PPISP-XGBoost reach 0.718 and 0.857, which are also higher than other methods. The results show that PPISP-XGBoost has the best predictive performance on dataset Dset_448.

Table 12 shows the prediction results of the nine prediction methods on the Dset_355 including PPISP-XGBoost. The ACC value of PPISP-XGBoost reaches 85.4%, which is 0.6%–5.1% higher than other methods. The MCC and F-Measure of PPISP-XGBoost reach 0.709 and 0.853, which are higher than other methods. The results show that PPISP-XGBoost has the best predictive performance on dataset Dset_355. We also included the comparison results for Dset_448 and Dset_355 under different prediction methods in Supplementary Fig. S4. It can be seen from Fig. S4 that each index for these two datasets also reached its maximum under the PPISP-XGBoost method.

In short, by comparing the prediction results with other methods on the four independent validation datasets, it is evident that PPISP-XGBoost shows superior robustness on different datasets. The above analysis has demonstrated that PPISP-XGBoost can, effectively, efficiently and accurately predict PPI sites, and has better generalizability than the other methods considered.

4. Conclusion

The study of PPI sites provides indispensable clues for understanding the biological processes involving proteins, and has immediate benefit towards a number of research areas, including drug design, construction of signal transduction networks, reconstruction of protein interaction networks, etc. Making good use of machine learning to identify and predict PPI sites can both effectively reduce experimental costs and improve prediction efficiency. This study proposed a new model, PPISP-XGBoost for the prediction of PPI sites. First, different features including: evolution information, sequence information, composition information, and physicochemical properties of the protein were used, then optimal features from these were extracted using four feature extraction algorithms: PsePSSM, PseAAC, hydropathy index and ASA. Next, minority classes were "oversampled" by the SMOTE method. By processing the extracted features using SMOTE and generating new balanced sample data, model overfitting caused by the reduction of the decision domain, can be effectively mitigated. Furthermore, the KPCA algorithm was used to dramatically reduce noise and redundancy in the multi-dimensional feature information. Finally, the obtained optimal feature vectors were used as input to the XGBoost classifier to predict PPI sites. When optimizing the loss function, XGBoost uses both first- and second-order derivatives, and adds a regularizing term to control the complexity of the model, which further alleviates the problem of over-fitting. Furthermore, the parallelization of the split point greatly increases the training rate. The experimental results show that the PPISP-XGBoost model achieved the best performance across a number of indicators in predicting PPI sites compared with other existing methods on five datasets. MCC and F-measure, are particularly significantly improved. Although the PPISP-XGBoost method has improved the accuracy of PPI sites prediction to a certain extent, we believe there is still room for further enhancement.

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

This work was supported by the National Natural Science Foundation of China (No. 61863010), the Key Research and Development Program of Shandong Province of China (No. 2019GGX101001), and the Key Laboratory Open Foundation of Hainan Province (No. JSKX202001).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compbiomed.2021.104516.

References

[1] H.X. Zhou, Y. Shan, Prediction of protein interaction sites from sequence profile and residue neighbor list, Proteins 44 (2001) 336–343.
[2] P.L. Gu, D.H. Morgan, M. Sattar, X.P. Xu, R. Wagner, M. Raviscioni, O. Lichtarge, A. J. Cooney, Evolutionary trace-based peptides identify a novel asymmetric interaction that mediates oligomerization in nuclear receptors, J. Biol. Chem. 280 (2005) 31818–31829.
[3] D. Dell'Orco, Fast predictions of thermodynamics and kinetics of protein-protein recognition from structures: from molecular design to systems biology, Mol. Biosyst. 5 (2009) 323–334.
[4] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, Y. Sakaki, A comprehensive two-hybrid analysis to explore the yeast protein interactome, Proc. Natl. Acad. Sci. Unit. States Am. 98 (2001) 4569–4574.
[5] Y. Ho, A. Gruhler, A. Heilbut, G.D. Bader, L. Moore, S.L. Adams, A. Millar, P. Taylor, K. Bennett, K. Boutilier, L.Y. Yang, C. Wolting, I. Donaldson, S. Schandorff, J. Shewnarane, M. Vo, J. Taggart, M. Goudreault, B. Muskat, C. Alfarano, D. Dewar, Z. Lin, K. Michalickova, A.R. Willems, H. Sassi, P.A. Nielsen, K.J. Rasmussen, J.R. Andersen, L.E. Johansen, L.H. Hansen, H. Jespersen, A. Podtelejnikov, E. Nielsen, J. Crawford, V. Poulsen, B.D. Sørensen, J. Matthiesen, R.C. Hendrickson, F. Gleeson, T. Pawson, M.F. Moran, D. Durocher, M. Mann, C.W. V. Hogue, D. Figeys, M. Tyers, Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry, Nature 415 (2002) 180–183.
[6] E.M. Marcotte, M. Pellegrini, H.L. Ng, D.W. Rice, T.O. Yeates, D. Eisenberg, Detecting protein function and protein-protein interactions from genome sequences, Science 285 (1999) 751–753.
[7] T.T. Aumentado-Armstrong, B. Istrate, R.A. Murgita, Algorithmic approaches to protein-protein interaction site prediction, Algorithm. Mol. Biol. 10 (2015) 7.
[8] Q. Zhang, Y.D. Zhang, S. Li, Y. Han, S.P. Jin, H.M. Gu, B. Yu, Accurate Prediction of Multi-Label Protein Subcellular Localization through Multi-View Feature Learning with RBRL Classifier, Brief. Bioinform, 2021. http://10.1093/bib/bbab012.
[9] Y. Murakami, K. Mizuguchi, Applying the Naïve Bayes classifier with kernel density estimation to the prediction of protein-protein interaction sites, Bioinformatics 26 (2010) 1841–1848.
[10] K. Dhole, G. Singh, P.P. Pai, S. Mondal, Sequence-based prediction of protein-protein interaction sites with L1-logreg classifier, J. Theor. Biol. 348 (2014) 47–54.
[11] B.Z. Zhang, J.Y. Li, L.J. Quan, Y. Chen, Q. Lü, Sequence-based prediction of protein-protein interaction sites by simplified long-short term memory network, Neurocomputing 357 (2019) 86–100.
[12] K.C. Chou, Prediction of protein cellular attributes using pseudo-amino acid composition, Proteins 43 (2001) 246–255.
[13] Q.W. Dong, X.L. Wang, L. Lin, Y. Guan, Exploiting residue-level and profile-level interface propensities for usage in binding sites prediction of proteins, BMC Bioinf. 8 (2007) 147.
[14] Z.Y. Xie, X.Y. Deng, K.X. Shu, Prediction of protein-protein interaction sites using convolutional neural network and improved data sets, Int. J. Mol. Sci. 21 (2020) 467.
[15] C. Chen, Q.M. Zhang, Q. Ma, B. Yu, LightGBM-PPI: predicting protein-protein interactions through LightGBM with multi-information fusion, Chemometr. Intell. Lab. Syst. 191 (2019) 54–64.
[16] J.C. Zeng, D.P. Li, Y.F. Wu, Q. Zou, X.R. Liu, An empirical study of features fusion techniques for protein-protein interaction prediction, Curr. Bioinf. 11 (2016) 4–12.
[17] Y.E. Göktepe, H. Kodaz, Prediction of protein-protein interactions using an effective sequence-based combined method, Neurocomputing 303 (2018) 68–74.
[18] B. Yu, C. Chen, X.L. Wang, Z.M. Yu, A.J. Ma, B.Q. Liu, Prediction of protein-protein interactions based on elastic net and deep forest, Expert Syst. Appl. 176 (2021) 114876.
[19] L. Nanni, S. Brahnam, A. Lumini, High performance set of PseAAC and sequence based descriptors for protein classification, J. Theor. Biol. 266 (2010) 1–10.
[20] Y.E. Göktepe, İ. İlhan, Ş. Kahramanlı, Predicting protein-protein interactions by weighted pseudo amino acid composition, Int. J. Data Min. Bioinf. 15 (2016) 272–290.
[21] Y. Xu, Z. Wang, C.H. Li, K.C. Chou, iPreny-PseAAC: identify C-terminal cysteine prenylation sites in proteins by incorporating two tiers of sequence couplings into PseAAC, Med. Chem. 13 (2017) 544–551.
[22] Y.J. Qi, Z. Bar-Joseph, J. Klein-Seetharaman, Evaluation of different biological data and computational classification methods for use in protein interaction prediction, Proteins 63 (2006) 490–500.
[23] M. Rashid, S. Ramasamy, G.P.S. Raghava, A simple approach for predicting protein-protein interactions, Curr. Protein Pept. Sci. 11 (2010) 589–600.
[24] S.M. Gomez, S.H. Lo, A. Rzhetsky, Probabilistic prediction of unknown metabolic and signal-transduction networks, Genetics 159 (2001) 1291–1298.
[25] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, Mach. Learn. 29 (1997) 131–163.
[26] M.H. Li, L. Lin, X.L. Wang, T. Liu, Protein-protein interaction site prediction based on conditional random fields, Bioinformatics 23 (2007) 597–604.
[27] G.T. Valente, M.L. Acencio, C. Martins, N. Lemke, The development of a universal in silico predictor of protein-protein interactions, PloS One 8 (2013), e65587.
[28] Q.Z. Hou, P.F.G.D. Geest, W.F. Vranken, J. Heringa, K.A. Feenstra, Seeing the trees through the forest: sequence-based homo- and heteromeric protein-protein interaction sites prediction using random forest, Bioinformatics 33 (2017) 1479–1487.
[29] Y. Ofran, B. Rost, Predicted protein-protein interaction sites from local sequence information, FEBS Lett. 544 (2003) 236–239.
[30] H. Neuvirth, R. Raz, G. Schreiber, ProMate: a structure based prediction program to identify the location of protein-protein binding sites, J. Mol. Biol. 338 (2004) 181–199.
[31] C.H. Yan, D. Dobbs, V. Honavar, A two-stage classifier for identification of protein-protein interface residues, Bioinformatics 20 (2004) i371–i378.
[32] Z.S. Wei, J.Y. Yang, H.B. Shen, D.J. Yu, A cascade random forests algorithm for predicting protein-protein interaction sites, IEEE Trans. NanoBioscience 14 (2015) 746–760.
[33] X.Y. Wang, B. Yu, A.J. Ma, C. Chen, B.Q. Liu, Q. Ma, Protein-protein interaction sites prediction by ensemble random forests with synthetic minority oversampling technique, Bioinformatics 35 (2019) 2395–2402.
[34] Z.S. Wei, K. Han, J.Y. Yang, H.B. Shen, D.J. Yu, Protein-protein interaction sites prediction by ensembling SVM and sample-weighted random forests, Neurocomputing 193 (2016) 201–212.
[35] M. Zeng, F. Zhang, F.X. Wu, Y. Li, J. Wang, M. Li, Protein-protein interaction site prediction through combining local and global features with deep neural networks, Bioinformatics 36 (2020) 1114–1120.
[36] N. Japkowicz, S. Stephen, The class imbalance problem: a systematic study, Intell. Data Anal. 6 (2002) 429–449.
[37] D.J. Yu, J. Hu, H. Yan, X.B. Yang, J.Y. Yang, H.B. Shen, Enhancing protein-vitamin binding residues prediction by multiple heterogeneous subspace SVMs ensemble, BMC Bioinf. 15 (2014) 297.
[38] K. Dhole, G. Singh, P. Pai, S. Mondal, SPRINGS: prediction of protein-protein interaction sites using artificial neural networks, J. Proteom. Comput. Biol. 1 (2014) 7.
[39] J. Zhang, L. Kurgan, SCRIBER: accurate and partner type-specifific prediction of protein-binding residues from proteins sequences, Bioinformatics 35 (2019) i343–i353.
[40] Y.W. Li, G.B. Golding, L. Ilie, DELPHI: accurate deep ensemble model for protein interaction sites prediction, Bioinformatics (2020). http://10.1093/bioinformatics/btaa750.
[41] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I. N. Shindyalov, P.E. Bourne, The protein data bank, Nucleic Acids Res. 28 (2000) 235–242.
[42] J.Y. Yang, A. Roy, Y. Zhang, BioLiP: a semi-manually curated database for biologically relevant ligand–protein interactions, Nucleic Acids Res. 41 (2013) D1096–D1103.
[43] S. Jones, J.M. Thornton, Analysis of protein-protein interaction sites using surface patches, J. Mol. Biol. 272 (1997) 121–132.
[44] P. Aloy, R.B. Russell, Interrogating protein interaction networks through structural biology, Proc. Natl. Acad. Sci. Unit. States Am. 99 (2002) 5896–5901.
[45] J. Mihel, M. Sikić, S. Tomić, B. Jeren, K. Vlahovicek, PSAIA-protein structure and interaction analyzer, BMC Struct. Biol. 8 (2008) 21.
[46] A. Dehzangi, R. Heffernan, A. Sharma, J. Lyons, K. Paliwal, A. Sattar, Gram-positive and Gram-negative protein subcellular localization by incorporating evolutionary-based descriptors into Chou's general PseAAC, J. Theor. Biol. 364 (2015) 284–294.
[47] H.B. Shen, K.C. Chou, Nuc-PLoc: a new web-server for predicting protein subnuclear localization by fusing PseAA composition and PsePSSM,, Protein Eng. Des. Sel. 20 (2007) 561–567.
[48] Q. Zhang, S. Li, B. Yu, Q.M. Zhang, Y. Han, Y. Zhang, Q. Ma, DMLDA-LocLIFT: identification of multi-label protein subcellular localization using DMLDA dimensionality reduction and LIFT classifier, Chemometr. Intell. Lab. Syst. 206 (2020) 104148.
[49] B. Yu, S. Li, W.Y. Qiu, M.H. Wang, J.W. Du, Y.S. 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.
[50] S.F. Altschul, E.M. Gertz, R. Agarwala, A.A. Schäffer, Y.K. Yu, PSI-BLAST pseudocounts and the minimum description length principle, Nucleic Acids Res. 37 (2009) 815–824.
[51] D.T. Jones, Protein secondary structure prediction based on position-specific scoring matrices, J. Mol. Biol. 292 (1999) 195–202.
[52] Q.M. Zhang, P.S. Liu, X. Wang, Y.Q. Zhang, Y. Han, B. Yu, StackPDB: predicting DNA-binding proteins based on XGB-RFE feature optimization and stacked ensemble classifier, Appl. Soft Comput. 99 (2021) 106921.
[53] B. Yu, C. Chen, H.Y. Zhou, B.Q. Liu, Q. Ma, GTB-PPI: predict protein–protein interactions based on L1-regularized logistic regression and gradient tree boosting, Genom. Proteom. Bioinf. (2021), https://doi.org/10.1016/j.gpb.2021.01.001.
[54] B. Yu, Z.M. Yu, A.J. Ma, C. Chen, B.Q. Liu, B.G. Tian, Q. Ma, DNNAce: prediction of prokaryote lysine acetylation sites through deep neural networks with multi-information fusion, Chemometr. Intell. Lab. Syst. 200 (2020) 103999.

[55] W.Y. Qiu, S. Li, X.W. Cui, Z.M. Yu, M.H. Wang, J.W. Du, Y.J. 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. 450 (2018) 86–103.
[56] C. Chen, Q.M. Zhang, B. Yu, Z.M. Yu, P.J. Lawrence, Q. Ma, Y. Zhang, Improving protein-protein interactions prediction accuracy using XGBoost feature selection and stacked ensemble classifier, Comput. Biol. Med. 123 (2020) 103899.
[57] J. Kyte, R.F. Doolittle, A simple method for displaying the hydropathic character of a protein, J. Mol. Biol. 157 (1982) 105–132.
[58] S.F. Altschul, T.L. Madden, A.A. Schäffer, J. Zhang, Z. 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.
[59] W.C. Wimley, S.H. White, Experimentally determined hydrophobicity scale for proteins at membrane interfaces, Nat. Struct. Biol. 3 (1996) 842–848.
[60] B. Lee, F.M. Richards, The interpretation of protein structures: estimation of static accessibility, J. Mol. Biol. 55 (1971) 379–400.
[61] R. Heffernan, Y.D. Yang, K. Paliwal, Y.Q. Zhou, Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility, Bioinformatics 33 (2017) 2842–2849.
[62] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.
[63] H. Shi, S.M. Liu, J.Q. 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.
[64] 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. 32 (2020) 13843–13862.
[65] 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, Chemometr. Intell. Lab. Syst. 197 (2020) 103919.
[66] B. Schölkopf, A. Smola, K.R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (1998) 1299–1319.
[67] M. Ringnér, What is principal component analysis, Nat. Biotechnol. 26 (2008) 303–304.
[68] T.Q. Chen, C. Guestrin, XGBoost: a calable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 785–794.
[69] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. 29 (2001) 1189–1232.
[70] 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.
[71] 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.
[72] Z.X. Zhao, H. Peng, C.W. Lan, Y. Zheng, L. Fang, J.Y. Li, Imbalance learning for the prediction of N6-Methylation sites in mRNAs, BMC Genom. 19 (2018) 574.
[73] Y. Sun, C.G. Castellano, M. Robinson, R. Adams, A.G. Rust, N. Davey, Using pre & post-processing methods to improve binding site predictions, Pattern Recogn. 42 (2009) 1949–1958.
[74] X.Y. Liu, J.X. Wu, Z.H. Zhou, Exploratory undersampling for class-imbalance learning, IEEE T. Syst. Man CY. B. 39 (2009) 539–550.
[75] J. Zhang, I. Mani, KNN approach to unbalanced data distributions: a case study involving information extraction, in: Proceedings of the ICML' 2003 Workshop on Learning from Imbalanced Datasets, 2003.
[76] M.A. Habib, F. Ibrahim, M.S. Mohktar, S.B. Kamaruzzaman, K.S. Lim, Recursive independent component analysis (ICA)-decomposition of ictal EEG to select the best ictal component for EEG source imaging, Clin. Neurophysiol. 131 (2020) 642–654.
[77] K. Kayabol, Approximate sparse multinomial logistic regression for classification, IEEE Trans. Pattern Anal. Mach. Intell. 42 (2020) 490–493.
[78] R. Tibshirani, Regression shrinkage and selection via the lasso: a retrospective, J. Roy. Stat. Soc. B. 73 (2011) 267–288.
[79] Y. Bengio, O. Delalleau, N.L. Roux, J.F. Paiement, P. Vincent, M. Ouimet, Learning eigenfunctions links spectral embedding and kernel PCA, Neural Comput. 16 (2004) 2197–2219.
[80] D.A. Engemann, A. Gramfort, Automated model selection in covariance estimation and spatial whitening of MEG and EEG signals, Neuroimage 108 (2015) 328–342.
[81] O.P. Tabbaa, C. Jayaprakash, Mutual information and the fidelity of response of gene regulatory models, Phys. Biol. 11 (2014), 046004.
[82] X.Q. Ru, L.D. Wang, L.H. Li, H. Ding, X.C. Ye, Q. Zou, Exploration of the correlation between GPCRs and drugs based on a learning to rank algorithm, Comput. Biol. Med. 119 (2020) 103660.
[83] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297.
[84] 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.
[85] S.R. Safavian, D. Landgrebe, A survey of decision tree classifier methodology, IEEE Trans. Syst. Man Cybern. 21 (1991) 660–674.
[86] S.K. Pal, S. Mitra, Multilayer perceptron, fuzzy sets, and classification, IEEE Trans. Neural Network. 3 (1992) 683–697.
[87] F. Nigsch, A. Bender, B.V. Buuren, J. Tissen, E. Nigsch, J.B.O. Mitchell, Melting point prediction employing k-nearest neighbor algorithms and genetic parameter optimization, J. Chem. Inf. Model. 46 (2006) 2412–2422.
[88] L. Breiman, Random forest, Mach. Learn. 45 (2001) 5–32.
[89] J. Fisher, Box, Guinness, gosset, Fisher, and small samples, Stat. Sci. 2 (1987) 45–52.