Neurocomputing 104 (2013) 180–190

Contents lists available at SciVerse ScienceDirect

Neurocomputing

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

Improving protein-ATP binding residues prediction by boosting SVMs with random under-sampling

Dong-Jun Yu [a,c], Jun Hu [a], Zhen-Min Tang [a], Hong-Bin Shen [b,*], Jian Yang [a], Jing-Yu Yang [a]

[a] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, PR China
[b] Department of Automation, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, PR China
[c] Changshu Institute, Nanjing University of Science and Technology, Changshu 215513, PR China

ARTICLE INFO

Article history:
Received 14 June 2012
Received in revised form
17 September 2012
Accepted 1 October 2012
Communicated by Dr. L. Kurgan
Available online 15 November 2012

Keywords:
Protein-ATP binding prediction
Position specific scoring matrix
Protein secondary structure
Random under-sampling
SVM ensemble
AdaBoost

ABSTRACT

Correctly localizing the protein-ATP binding residues is valuable for both basic experimental biology and drug discovery studies. Protein-ATP binding residues prediction is a typical imbalanced learning problem as the size of minority class (binding residues) is far less than that of majority class (non-binding residues) in the entire sequence. Directly applying the traditional machine learning approach for this task is not suitable as the learning results will be severely biased towards the majority class. To circumvent this problem, a modified AdaBoost ensemble scheme based on random under-sampling is developed. In addition, effectiveness of different features for protein-ATP binding residues prediction is systematically analyzed and a method for objectively reporting evaluation results under the imbalanced learning scenario is also discussed. Experimental results on three benchmark datasets show that the proposed method achieves higher prediction accuracy. The proposed method, called TargetATP, has been implemented with Java programming language and is distributed via Java Web Start technology. TargetATP and the datasets used are freely available at http://www.csbio.sjtu.edu.cn/bioinf/targetATP/ for academic use.

1. Introduction

Adenosine-5′-triphosphate (ATP) is an important molecule in cell biology which plays an important role in membrane transport, cellular motility, muscle contraction, signaling, replication and transcription of DNA, and various metabolic processes [1–3]. ATP interacts with protein through protein-ATP binding residues and provides chemical energy to protein through the hydrolysis of ATP. Powered by the chemical energy, a protein can perform various biological functions. Naturally, ATP needs to interact with large number of proteins during cellular activities to accomplish its tasks, which make it necessary to investigate protein-ATP binding residues. In addition, the ATP binding residues can also be exploited as valuable targets for chemo-therapeutic agents [2]. Hence, accurately localizing the protein-ATP binding residues is of significant importance for both protein function analysis and drug design.

Although much progress has been made in identifying the protein-ATP interaction residues, the traditional experimental methods are often suffered from difficulties of lab intensive, expensive, and time-consuming, and it is highly desired to develop intelligent computational methods for protein-ATP binding residues prediction [3]. Nobeli et al. [4] performed pioneer work on molecular recognition and discrimination of adenine and guanine ligand moieties in complexes with proteins. Later on, predictors for identifying ATP-binding residues were developed successively. To name a few of them: ATPint [5] was the first custom-designed protein-ATP binding residues predictor built on a dataset consisting of 168 non-redundant ATP binding proteins. In ATPint, features derived from position specific scoring matrix (PSSM) and several other sequential descriptors were used for prediction. Recently, Kurgan et al. developed two more accurate predictors called ATPsite [6] and NsitePred [7] based on a much larger dataset, which is composed of 227 non-redundant ATP binding proteins, and better prediction results were achieved.

From the perspective of machine learning, protein-ATP binding residues prediction is a typical imbalanced learning problem, among which the number of samples in different classes differs significantly. For example, in ATP168 (refer to Section 2) dataset, the number of the majority class (non-binding residues) is more than 11 times of that of the minority class (binding residues). Directly applying the traditional machine learning algorithms, which assume that samples in different classes are balanced, often leads to a poor performance. The basic solution to imbalanced learning is the sample rescaling strategy [8] that tires to

* Corresponding author.
E-mail address: hbshen@sjtu.edu.cn (H.-B. Shen).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.10.012

balance data by changing the distribution of samples in different classes. Over-sampling [9] and under-sampling [10] are the two most commonly used implementations of the sample rescaling strategy. Over-sampling increases the size of a minority class by synthesizing new samples from or just directly replicating the randomly selected samples; while under-sampling alters the size of a majority class by removing samples from the original dataset. Until now, no evidences or theoretical justifications can show that over-sampling prevails over under-sampling and vise versa [11]. The defects of over-sampling lie in two aspects: on the one hand, over-sampling enlarges the training dataset which may increase the subsequent training and predicting time; on the other hand, as over-sampling simply appends replicated samples to the original dataset, multiple instances of certain examples become "tied," leading to over-fitting problem [12]. Compared with over-sampling, under-sampling can provides a much smaller and compact training dataset since it removes samples from the original dataset. At the same time, it can be imagined that part of the important pertaining to the majority class could be lost in the under-sapling process, and this may deteriorate the classifier's performance to some extent.

Previous studies [13] have shown that classifier ensemble is a promising route to relieve the impact of information loss caused by under-sampling. In this study, we developed TargetATP, a method which combines multiple under-samplings with classifier ensemble to further improve the accuracy of protein-ATP binding residues prediction: first, we sample several different majority training subsets by random under-sampling the majority class several times; then, we train a base classifier, i.e., SVMs in this study, on each of the majority training subsets plus the minority training set; finally, the trained base classifiers are ensembled to perform the final decision.

2. Material and methods

2.1. Benchmark datasets

In this study, three main benchmark datasets were used to demonstrate the efficacy of the proposed method. The first dataset, which was collected by Chauhan et al. [5], consists of 168 non-redundant protein sequences, denoted as ATP168, among which there exists 3,056 ATP-binding residues. The second dataset constructed by Chen et al. [6] consists of 227 non-redundant protein sequences, denoted as ATP227, among which there exists 3,393 ATP-binding residues. The sequence identity of any two proteins in both above two benchmark datasets is less than 40%. The third dataset was constructed by Firoz et al. [14] and consists of 131 protein sequences, denoted as ATP131, which has been removed redundancy at 30% sequence identity cutoff. In addition, in order to blindly test the performance of our predictor TargetATP, the same independent dataset as in [6], which contains 17 proteins that bind to ATP, was also utilized to evaluate the generalization ability of the proposed method.

2.2. Feature extraction

2.2.1. Position specific scoring matrix

Position specific scoring matrix (PSSM) well encodes the evolutionary information of a protein sequence. Tremendous previous studies have shown its prominent discriminative capability for many prediction problems in bioinformatics, such as protein function prediction [15], protein-ATP binding sites prediction [16], transmembrane helices prediction [17], protein secondary structure prediction [18], and subcellular localization prediction [19,20], etc.

For a protein sequence P with n amino acid residues, we obtain its PSSM (n rows and 20 columns) by using the PSI-BLAST [21] to search the Swiss-Prot database through three iterations with 0.001 as the E-value cutoff for multiple sequence alignment against the sequence of the protein. Then, we normalize the obtained PSSM by the logistic function defined as follows:

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

where x is the original score in PSSM matrix.

The sliding window technique was used to extract the PSSM-based feature vector of a residue, i.e., whether a residue belongs to a protein-ATP binding residue or not is predicted based on the PSSM scores of its neighboring residues with a window centered at the residue. In this study, the size of sliding window is set to be 17 and thus the dimensionality of the obtained PSSM-based feature vector, denoted as *LogisticPSSM* feature, is $17\times 20=340$.

2.2.2. Protein secondary structure

Previous studies have shown that the protein secondary structures are partially relevant to the ATP-binding residues, thus appropriately utilizing protein secondary structure information is helpful to improve the performance of ATP-binding residues prediction [6]. In this study, for a given amino acid sequence, we obtain its predicted secondary structure information by applying PSIPRED [22], which predicts the probabilities belonging to three secondary structure classes (coil (C), helix (H), and strand (E)) of each residue. Thus, for a protein with n residues, we obtain a $n\times 3$ probability matrix, which represents the predicted secondary structure information of the protein. Again, a sliding window with size 17 was used to extract protein-secondary-structure-based feature, denoted as *PSS*, of each reside and the dimensionality of the extracted feature is $17\times 3=51$.

By carefully observing Fig. 1 of the predicted protein secondary structural compositions in the two benchmark datasets of ATP168 and ATP227, two observations can be drawn:

First, the secondary structure composition in binding residues is different from that in non-binding residues. For example, in ATP168, the percentages of coil (C), helix (H), and strand (E) in binding residues are 49.6%, 26.0%, and 24.4%, respectively; while those in non-binding residues are 41.7%, 40.3%, and 18.0%, respectively.

Second, the secondary structure composition has specific composition mode in both the binging residues and the non-binding residues and is consistent in two benchmark datasets. Taking secondary structure composition in non-binding residues as example, the percentages of coil (C), helix (H), and strand (E) in ATP168 are 41.7%, 40.3%, and 18.0%, respectively; while those in

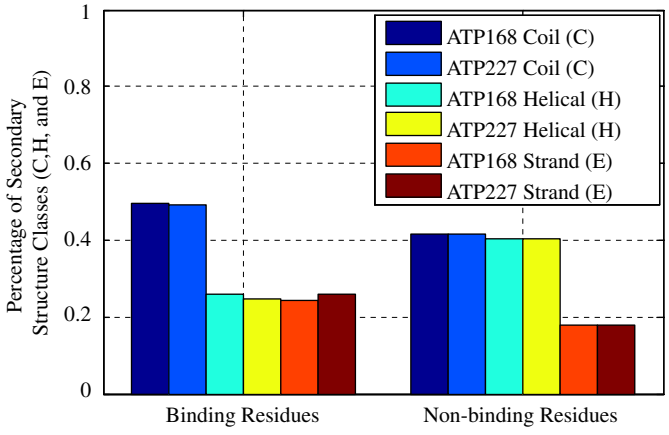


Fig. 1. Protein secondary structure composition comparison of binding residues and non-binding residues in ATP168 and ATP227.

ATP227 are 41.7%, 40.4%, and 17.0%, respectively. The secondary structure composition of non-binding residues in the two datasets is almost the same and the biggest difference (i.e., percentages of strand (E)) is only about 1%.

The above two observations show that protein secondary structure information can be also a useful discriminator that can help to improve protein-ATP binding residues prediction accuracy.

It has not escaped our notice that other feature encoding system based on the amino acid physiochemical characteristics [5,6], such as hydrophobicity, beta-sheet, polarity, solvation potential, and net charge, etc., have also been tried for protein-ATP binding residues. However, our preliminary tests show that incorporating these physiochemical features into PSSM and PSS based features only slightly affects the prediction results, and may even deteriorate the prediction performance in some cases. Thus, in current study, we only incorporate the two features of PSSM and PSS for constructing the predictor.

2.3. Multiple under-samplings and SVM ensemble

As described in Section 1, random under-sampling can effectively balance the samples in different classes and provide a much smaller training dataset, consequently speeding up the training and prediction processes. However, random under-sampling may also lose information carried by those non-sampled points and thus potentially deteriorates the prediction performance of a prediction model. To circumvent this problem, a feasible route is to combine under-sampling technique with classifier ensemble. More specifically, we randomly sample the majority class (non-binding residues in this study) L times (in the presented study, $L=5$) with under-sampling technique and thus obtained L majority training subsets. The L majority training subsets each plus the minority training set constitute L new training datasets. Then, a kind of machine learning model is trained on each of the L new training datasets; in the prediction stage, for each residue in a given protein sequence, its probability of belonging to the binding residue class is predicted by each of the L machine learning models; finally, the L probabilities of the residue belonging to the binding residue class are fused by an appropriate classifier ensemble strategy. By doing so, on one hand, merits derived from under-sampling such as sample balance can be retained; on the other hand, multiple samplings can reduce the loss of information caused by random under-sampling to some extent, thus may potentially provide better prediction performance. In this paper, we use the support vector machine (SVM) as the base classifier for constructing the ensemble.

Support vector machine (SVM) was first proposed by Vapnik [23]. Recently, SVM has been broadly investigated in a wide variety of bioinformatics areas and great success has been made. Different from the traditional pattern recognition techniques (e.g. neural networks) that are based on the minimization of the empirical risk, the SVM minimizes the structural risk. In this study, LIBSVM [24], which is freely available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/, was used. Here, radial basis function is chosen as the kernel function. The other two parameters, i.e., the regularization parameter γ and the kernel width parameter σ, are optimized based on ten-fold cross-validation using a grid search strategy in the LIBSVM software.

As to classifier ensemble, Xu et al. categorized it into three Levels (Level 1, Level 2, and Level 3) according to the output types of the base classifiers [25]: in Level 1 classifier ensemble, each base classifier outputs an abstract label; while in Level 2 classifier ensemble, each base classifier outputs a subset of ranked labels; in Level 3 classifier ensemble, each base classifier outputs a vector of measurement with each on how likely a label is. In fact, Level 1 and 2 can be regarded as the special cases of Level 3. Details for three Levels please refer to [25]. In the present study, for a residue in a protein to be predicted, each SVM (base classifier) outputs a 2D vector, the elements among which measure probabilities of the residue for being binding or non-binding residues, respectively. Thus, ensemble these multiple SVMs falls into Level 3 defined by Xu et al. [25].

Classifier ensemble has been widely applied in bioinformatics, such as protein fold prediction [26,27], protein subcellular localization prediction [19], and protein structural class prediction [28], etc. Previous studies have shown that different ensemble results can be obtained by applying different ensemble schemes [29–33]. In [34], Kuncheva well surveyed many widely used ensemble schemes and pointed out that different ensemble schemes have their own merits and shortcomings, and there does not exist a general "best" ensemble scheme for all kinds of applications. For a specific application, e.g. protein-ATP binding residues prediction in this study, people can try to choose a suitable ensemble scheme, but the theoretical justification for the best choice is still hard to be derived. Considering this, in this study, we tested four popular ensemble schemes [34] including *Maximum*, *Minimum*, *Mean*, and *MAdaBoost*. Note that a modified *AdaBoost* (*MAdaBoost*) rather than the traditional *AdaBoost* was developed to facilitate the protein-ATP binding residues prediction.

Let $\Omega=\{\omega_1,\omega_2,\cdots,\omega_C\}$ be the set of classes, $\Gamma=\{S_1,S_2,\cdots,S_L\}$ be the set of L base classifiers (SVMs in this study), and $X^{\mathrm{Tr}}=\{\mathbf{x}_t^{\mathrm{Tr}}\}_{t=1}^{N}$ be the training dataset. Each base classifier S_i outputs a C-dimensional vector $(s_{i,1}(\mathbf{x}),s_{i,2}(\mathbf{x}),\cdots,s_{i,C}(\mathbf{x}))^T$, where $s_{i,j}(\mathbf{x})$ measures the probability of sample $\mathbf{x}$ being classified into class j, $1\le j\le C$ (Level 3, as described above). The outputs of the L base classifiers for sample $\mathbf{x}$ constitute a decision profile, denoted as $DP(\mathbf{x})$, as follows:

$$DP(\mathbf{x})=\begin{bmatrix} s_{1,1}(\mathbf{x}) & \cdots & s_{1,j}(\mathbf{x}) & \cdots & s_{1,C}(\mathbf{x}) \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ s_{i,1}(\mathbf{x}) & \cdots & s_{i,j}(\mathbf{x}) & \cdots & s_{i,C}(\mathbf{x}) \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ s_{L,1}(\mathbf{x}) & \cdots & s_{L,j}(\mathbf{x}) & \cdots & s_{L,C}(\mathbf{x}) \end{bmatrix} \tag{2}$$

By applying an ensemble scheme, the ensembled classifier, denoted as *EnsembledClassifier*, can be obtained:

$$EnsembledClassifier = EnsembleScheme(\{S_1,S_2,\cdots,S_L\},X^{\mathrm{Tr}}) \tag{3}$$

Then, for a testing sample $\mathbf{x}\in X$, the ensembled classification result, denoted as $\mu(\mathbf{x})$, can be obtained by applying an ensemble scheme as follows:

$$\mu(\mathbf{x}) = EnsembledClassifier(\mathbf{x}) \tag{4}$$

where $\mu(\mathbf{x})=(\mu_1(\mathbf{x}),\mu_2(\mathbf{x}),\cdots,\mu_j(\mathbf{x}),\cdots,\mu_C(\mathbf{x}))^T$, and $\mu_j(\mathbf{x})$ is the confidence of $\mathbf{x}$ being classified into class ω_j after classifier ensemble.

(A) *Maximum* Ensemble

$$\mu_j(\mathbf{x})=\max_{1\le i\le L} s_{i,j}(\mathbf{x}) \tag{5}$$

(B) *Minimum* Ensemble

$$\mu_j(\mathbf{x})=\min_{1\le i\le L} s_{i,j}(\mathbf{x}) \tag{6}$$

(C) *Mean* Ensemble

$$\mu_j(\mathbf{x})=\frac{1}{L}\sum_{i=1}^{L} s_{i,j}(\mathbf{x}) \tag{7}$$

(D) *MAdaBoost* Ensemble

The idea of *AdaBoost* [35] is to develop the classifier team S incrementally, adding one classifier each time. The base classifier

Input	Training dataset $X^{Tr}=\{\mathbf{x}_t^{Tr}\}_{t=1}^{N}$, Independent evaluation dataset $X^{Eval}=\{\mathbf{x}_t^{Eval}\}_{t=1}^{M}$. L - the number of base classifiers.	
Output	The 2-tuple $\left(S,\{\varepsilon_i\}_{i=1}^{L}\right)$, where S is the ensembled classifier and ε_i is the corresponding weighted ensemble error of the i-th base classifier.	Notice: Here is an error. The last classifier number may be not equal to L, since the step 2.5.
1	**Initialization**	
	1.1 Set the weights $\mathbf{W}^1=\left(w_1^1,\cdots,w_M^1\right)$, $w_t^1\in[0,1]$, $\sum_{t=1}^{M}w_t^1=1$.(Usually $w_t^1=1/M$)	
	1.2 Initialize the ensemble $S=\varnothing$	
2	**Training**	
	2.1 For $i=1,\cdots,L$	
	2.2 Build a training subset X_i^{Tr} from dataset X^{Tr} by random under-sampling the majority classes	
	2.3 Train a classifier S_i on X_i^{Tr}	
	2.4 Calculate the weighted ensemble error at step i by $\varepsilon_i=\sum_{t=1}^{M}w_t^i l_t^i$, where $l_t^i=1$ if S_i misclassifies $\mathbf{x}_t^{Eval}$ and $l_t^i=0$ otherwise.	
	2.5 If $\varepsilon_i=0$ or $\varepsilon_i\geq 0.5$	
	2.5 Ignore S_i and reinitialize the weights w_t^i to $1/M$ and continue.	Step 2.5: We suggest that the main reason to make the error bigger than 0.5 is that the sample weights are unreasonable. So, we reinitialize the sample weights.
	2.6 Else	
	2.7 Calculate $\beta_i=\dfrac{\varepsilon_i}{1-\varepsilon_i}$	
	2.8 Update the individual weights: $w_t^{i+1}=\dfrac{w_t^i\beta_i^{1-l_t^i}}{\sum_{t=1}^{n}w_t^i\beta_i^{1-l_t^i}}, t=1,\ldots,M$	
	2.9 $S=S\cup\{S_i\}$	
	2.10 End If	
	2.11 End For	
	2.12 Return S and $\{\varepsilon_i\}_{i=1}^{L}$	

Fig. 2. Algorithm of *MAdaBoost*.

S_i that joins the ensemble at step i is trained on a training subset selectively sampled from the training dataset X^{Tr} by applying a sample-distribution-based sampling technique. As we try to systematically investigate the effectiveness of the random under-sampling technique on protein-ATP binding residues prediction, we will not directly apply the traditional *AdaBoost* algorithm. Instead, a modified *AdaBoost* algorithm, i.e., *MAdaBoost*, which combines the idea of *AdaBoost* and random under-sampling, is used.

In *MAdaBoost*, random under-sampling is used to construct training subsets as opposed to the sample-distribution-based sampling technique used in the traditional *AdaBoost* [35] during the ensemble procedure. Another difference is that in the traditional *AdaBoost*, samples in the whole training dataset are used as evaluation samples to evaluate the ensemble error of each base classifier; while in *MAdaBoost*, samples in an independent evaluation dataset are used. The reason for using an independent evaluation dataset is to guarantee that the evaluation samples and the samples chosen to train base classifier do not originate from the same protein sequence. Thus, the inputs of *MAdaBoost* algorithm are training dataset $X^{\mathrm{Tr}}=\{\mathbf{x}_t^{\mathrm{Tr}}\}_{t=1}^{N}$, evaluation dataset $X^{Eval}=\{\mathbf{x}_t^{Eval}\}_{t=1}^{M}$, and L-the number of base classifiers. Fig. 2 presents the detailed procedure of the *MAdaBoost*. When applying the *MAdaBoost*, in each round of the K-fold cross-validation (for details of cross-validation, refer to Section 3.2), one subset was used for testing; among the remaining $K-1$ subsets, one subset was used to construct X^{Eval} and other K-2 subsets were used to construct X^{Tr}; this practice continued until all the K subsets of the dataset were traversed over;

Then, for an unseen sample $\mathbf{x}$, its support for class ω_j $(1\leq j\leq C)$ obtained from the ensembled classifier $(S,\{\varepsilon_i\}_{i=1}^{L})$ is formulated as

$$\mu_j(\mathbf{x})=\sum_{S_i(\mathbf{x})=\omega_j}(1-\varepsilon_i)\times s_{i,j}(\mathbf{x})+\sum_{S_i(\mathbf{x})\neq\omega_j}\varepsilon_i\times s_{i,j}(\mathbf{x}) \tag{8}$$

where ε_i is the weighted ensemble error of classifier i.

Finally, the predicted class (in this case residue is binding or non-binding ATP) is selected based on the threshold on probability, i.e., all residues with probability above threshold are marked as ATP binding. How to choose the threshold T for reporting will be discussed in Section 3.1.

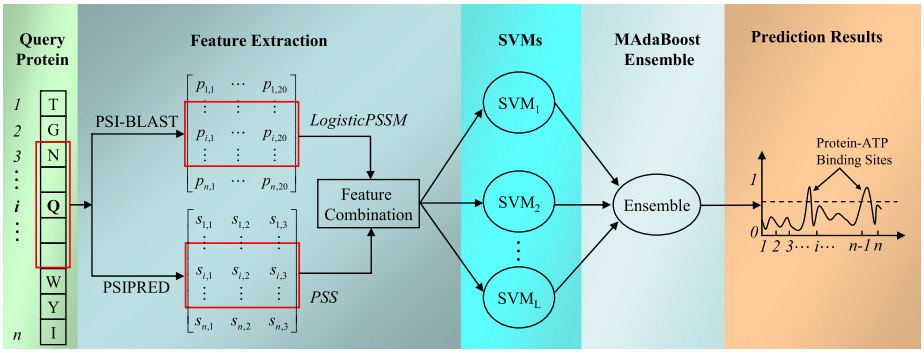


Fig. 3. Architecture of the TargetATP.

2.4. Architecture of the TargetATP

Fig. 3 illustrates the architecture of the proposed TargetATP for protein-ATP binding residues prediction. For a protein to be predicted, the TargetATP first extract *LogisticPSSM* and *PSS* features of each residue by calling the PSI-Blast and PSIPRED programs and applying sliding window technique; then, the two extracted features are combined and fed to L individual SVMs; the outputs of L SVMs are ensembled by applying an appropriate ensemble scheme; finally, a threshold is used to determine whether the residue is a ATP binding residue.

3. Performance evaluation

3.1. Evaluation indexes

In this study, four routinely used evaluation indexes in this filed, i.e., *Specificity* (*Spe*), *Sensitivity* (*Sen*), *Accuracy* (*Acc*), and the Matthews correlation coefficient (*MCC*) were taken to evaluate the performance of the proposed method as defined:

$$Specificity = \frac{TN}{TN+FP} \tag{9}$$

$$Sensitivity = \frac{TP}{TP+FN} \tag{10}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{11}$$

$$MCC = \frac{TP\cdot TN - FP\cdot FN}{\sqrt{(TP+FP)\cdot(TP+FN)\cdot(TN+FP)\cdot(TN+FN)}} \tag{12}$$

where *TP*, *FP*, *TN*, and *FN* are the abbreviations of true positive, false positive, true negative, and false negative, respectively. Note that these evaluation indexes are threshold-dependent and how to objectively report them will be further discussed. We also exploited another evaluation index *AUC*, which is the area under the Receiver Operating Characteristic (*ROC*) curve. The *AUC* is threshold independent and increases in direct proportion to the prediction performance.

For a soft-type classifier as in this study, i.e., classifier that output a continuous numeric value to represent the confidence of a sample belonging to the predicted class, gradually adjusting classification threshold will produce a series of confusion matrices [36]. From each confusion matrix, the corresponding *Spe*, *Sen*, *Acc*, and *MCC* can then be computed. In other words, these four evaluation indexes are threshold dependent, i.e., the values of them vary with the threshold chosen. Thus, how to objectively report these evaluation indexes is an important problem, especially in the situation of imbalanced learning scenario, where the numbers of samples in different classes are significantly unequal.

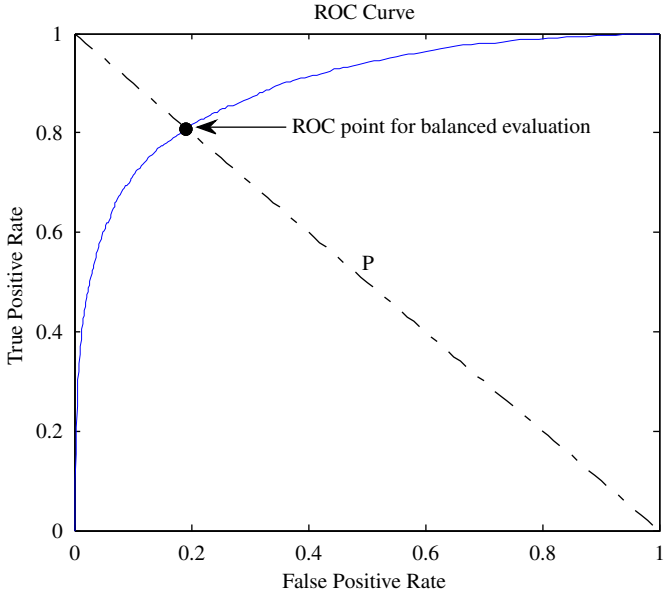


Fig. 4. Illustration of determining the ROC point on a ROC Curve for balanced evaluation.

Under the imbalanced learning scenario, over pursuing the overall accuracy is not appropriate and can be deceiving for evaluating the performance of a predictor/classifier. In general, people would expect that a predictor/classifier can provide high accuracy of the minority class (e.g. binding residues in this study) without severely jeopardizing the accuracy of the majority class [36] (non-binding residues). In light of this, the threshold, by which the value of the *false positive rate* ($FPR=FP/(FP+TN)$) is equal to that of the *false negative rate* ($FNR=FN/(FN+TP)$) [37], was chosen to report those threshold-dependent evaluation indexes. For the convenience of latter description, here we define concepts of *balanced evaluation* and *imbalanced evaluation* as follows.

3.1.1. Balanced evaluation

Those threshold-dependent evaluation indexes are reported with a threshold, by which the value of the *false positive rate* is equal to that of the *false negative rate*.

3.1.2. Imbalanced evaluation

Any threshold except the one for balanced evaluation can be used for imbalanced evaluation, where the *false positive rate* and the *false negative rate* are different.

Then, an issue should be clearly addressed here is how to find the threshold for balanced evaluation. As described above, for a soft-type classifier, gradually adjusting threshold will produce a series of confusion matrices. From each confusion matrix, a ROC point, whose coordinate is $(FP/(FP+TN),\ TP/(FN+TP))$, can be calculated. A series of ROC points constitute the ROC curve. Fig. 4 illustrates an exemplary ROC curve. As each threshold corresponds to a ROC point, thus the problem of how to find the threshold for balanced evaluation is equal to which ROC point should be used for balanced evaluation. We argue that the ROC point for balanced evaluation is the intersection point (black circle in Fig. 4) of the ROC curve and the line P which passes through the points (0, 1) and (1, 0) as shown in Fig. 4. The proof can be found in Appendix A.

3.2. Cross-validation

In this study, K-fold cross-validation was performed on all the three benchmark datasets for evaluating the performances. In the present study, our purpose is to predict whether a residue is a binding residue or not. If applying traditional cross-validation procedure, residues in all the training protein sequences are randomly partitioned into K disjoint subsets; then, one subset was used for testing and remaining $K-1$ subsets were used for training; this practice continued until all the K subsets of the dataset were traversed over.

However, we believe that directly applying the above cross-validation procedure is not appropriate because the testing residues and training residues may originate from the same protein sequence. In light of this, we would rather perform cross-validation on a higher level, i.e., on protein sequences. More specifically, training protein sequences are first randomly partitioned into K disjoint subsets; then, one subset was used for testing and remaining $K-1$ subsets were used for training; this practice continued until all the K subsets of the dataset were traversed over; the final performance was obtained by averaging the performance of all the K testing subsets.

4. Experimental results

4.1. Performances on different kinds of PSSM-based variation features

In this study, we have extracted the *LogisticPSSM* feature of a residue for protein-ATP binding residues prediction as described in Section 2.2. However, there also exist many other methods for extracting PSSM-based features, which have been used in different tasks. In light of this, we have also exploited four other PSSM-based feature extraction methods based on sliding window technique as listed in Table 1 for comparisons and tried to find a best PSSM-based feature encoding system for protein-ATP binding residues prediction.

Table 1
Five PSSM-based features with sliding window size $W=17$.

Feature types	Dimensionality
OriginalPSSM	340
LogisticPSSM (this paper)	340
tPSSM [6]	180
AveragePSSM [38]	20
FilteredPSSM [19]	20

OriginalPSSM

The method for extracting *OriginalPSSM* feature is the same as that for extracting *LogisticPSSM* feature except that the former is based on raw PSSM scores, while the latter is based on a PSSM normalized by the logistic function as defined in Eq. (1).

tPSSM [6]

First, the raw PSSM scores are transformed by applying the following function:

$$f(x)=\frac{1}{1+2^{-x}} \tag{13}$$

where x is the original score from the PSSM profile; then, for a size W sliding window centered at the i-th residue, a $W\times 20$ feature matrix for the residue is obtained; thereafter, the top and button rows of the feature matrix are averaged and a $(((W-1)/2)+1)\times 20$ feature matrix is obtained; finally, the $(((W-1)/2)+1)\times 20$ feature matrix is converted to a feature vector.

AveragePSSM [38]

The 20-D *AveragePSSM* feature of a residue is obtained by applying the sliding widow technique on a PSSM and averaging the residue's $W\times 20$ feature matrix by row.

FilteredPSSM [19]

It has been widely accepted that a positive score in the PSSM means that the corresponding mutation occurs more frequently in the alignment than expected by chance, while a negative score one means just the opposite. In light of this view, to enhance the contribution of evolutionary conserved positions, the PSSM is thus filtered by retaining only the residues whose scores at a given position are higher than a predefined threshold [19]. Then, a sliding window is applied to the filtered PSSM and the 20-D *FilteredPSSM* feature of each residue is obtained by averaging the $W\times 20$ feature matrix by row. Similar to [19], two thresholds, i.e., 0.33 and 0.75, were tried and the better one (0.33) was used for reporting.

Table 2 illustrates the discriminative performance comparison of the five PSSM-based features on ATP168 and ATP227 over five-fold cross-validation with single SVM classifier (no ensemble) under *balanced evaluation*. Note that in the present experiments, the negative samples (non-binding residues) were balanced to the positive samples (binding residues) by applying a random under-sampling technique. To avoid the potentially biased results caused by random under-sampling, for each type of the five PSSM-based features, we independently performed experiments 10 times and the averaged results were then reported.

By carefully analyzing Table 2, several insights can be obtained:

First, appropriately normalizing PSSM scores does help to improve the discriminative capability. For example, the *AUC* for the *LogisticPSSM* feature are 0.854 and 0.877, which are about 1.5% and 1.3% higher than that for the *OriginalPSSM* feature on ATP168 and ATP227, respectively. As to other three evaluation

Table 2
Discriminative performance comparison of the five PSSM-based features on ATP168 and ATP227 over five-fold cross-validation with single SVM classifier (no ensemble) under balanced evaluation.

Dataset	Feature types	Sen (%)	Spe (%)	Acc (%)	AUC
ATP168	OriginalPSSM [39]	75.6	76.2	76.2	0.839
	tPSSM [6]	75.5	75.8	75.8	0.834
	AveragePSSM [38]	69.6	69.9	69.9	0.760
	FilteredPSSM[19]	69.3	69.7	69.6	0.759
	LogisticPSSM	77.2	77.4	77.4	0.854
ATP227	OriginalPSSM [39]	78.9	79.2	79.2	0.864
	tPSSM [6]	77.9	78.1	78.1	0.857
	AveragePSSM [38]	70.8	71.1	71.1	0.779
	FilteredPSSM[19]	70.4	70.7	70.7	0.775
	LogisticPSSM	79.9	80.2	80.1	0.877

indexes, i.e., *Sen*, *Spe*, and *Acc*, the *LogisticPSSM* feature also outperforms the *OriginalPSSM* feature, and these improvements are consistent on both benchmark datasets.

Second, different PSSM-based features have different discriminative capabilities for specific protein function prediction. Taking protein-ATP binding residues prediction investigated in this study as an example, we found that the *LogisticPSSM* feature is much better than other four PSSM-based features, at least on the tested two benchmark datasets.

From another perspective, *tPSSM* [6], *AveragePSSM* [38], and *FilteredPSSM* [19] features can be considered as the derivatives of the *OriginalPSSM* feature, i.e., features further extracted from the *OriginalPSSM* feature. However, these deliberately extracted features do not provide better discriminative capabilities in this study. As shown in Table 2, the *OriginalPSSM* feature significantly outperforms the *AveragePSSM* and *FilteredPSSM* features, and is even better than the *tPSSM* feature that has been used in the ATPsite predictor [6]. We speculate that although the *tPSSM*, *AveragePSSM* and *FilteredPSSM* features derived from the *OriginalPSSM* may provide low-dimensional feature vectors, thus subsequently accelerates the training and predicting processes, some useful information buried in the *OriginalPSSM* is lost at the same time that deteriorates the discriminative capability. It is worth pointing out that different results were obtained when we applied the same *tPSSM* encoding system on the same ATP227 dataset as in [6], i.e., an *AUC* of 0.857 was obtained in this study as shown in Table 2, while 0.824 reported in [6]. The following two factors may account for this improvement: (1) In our experiments, a most recently released Swiss-Prot database composed of more protein sequences was used for performing PSI-BLAST search, thus can provide more accurate evolutionary information of a protein; (2) When the kernel function is chosen, the performance of SVM is significantly affected by the regularization parameter γ and the kernel width parameterσ. Considering this, we used a two stage solution to optimize these two parameters as much as possible: first, the intervals of γ and σ for grid search were preliminarily determined by trail and error; second, the grid search technique was performed on intervals of γ and σ for further optimization.

4.2. Combining PSS with LogisticPSSM

As we have shown above that *PSS* features are statistically discriminative features for classifying ATP-binding residues, thus appropriately incorporating *PSS* information into *LogisticPSSM* feature is expected to further improve the performance of ATP-binding residues prediction. In this study, a 391-D feature vector is obtained by combing the 340-D *LogisticPSSM* feature with the 51-D *PSS* feature as defined in Section 2.2 and is used as the input of the prediction model. Table 3 compares the performance of *LogisticPSSM+PSS* with that of *LogisticPSSM* on the two benchmark datasets over five-fold cross-validation with single SVM classifier (no ensemble) under balanced evaluation. From Table 3, it is easy to find that the prediction performances are indeed improved on both tested datasets, although not significant, by combining the two features.

The results listed in Table 3 were obtained by utilizing random under-sampling technique. As stated in previous section, directly applying the traditional machine learning approach under imbalanced learning scenario is not suitable as the learning results will be biased towards the majority class and thus lead to an inferior performance. To demonstrate this point, we also performed experiments on the two benchmark datasets not using random under-sampling technique. The *AUC*s for predictors of using random under-sampling on ATP168 and ATP227 are 0.859 and 0.882 (refer to Table 3), respectively, which are consistently better than 0.843 and 0.871 for the predictors trained on the whole dataset derived from ATP168 and ATP227, respectively. In addition, another advantage of random under-sampling is that it can help to significantly reduce computation time and storage cost, which is especially useful in many large-scale real-world imbalanced learning applications.

Table 3
Performance comparison between *LogisticPSSM+PSS* and *LogisticPSSM* on ATP168 and ATP227 with single SVM classifier (no ensemble) under balanced evaluation.

Dataset	Feature	*Sen* (%)	*Spe* (%)	*Acc* (%)	*AUC*
ATP168	*LogisticPSSM*	77.2	77.4	77.4	0.854
	LogisticPSSM+PSS	77.6	77.9	77.9	0.859
ATP227	*LogisticPSSM*	79.9	80.2	80.1	0.877
	LogisticPSSM+PSS	80.1	80.2	80.2	0.882

Table 4
Performance comparison of different ensemble strategies with *LogisticPSSM+PSS* feature on ATP168 and ATP227 over five-fold cross-validation under balanced evaluation.

Dataset	Ensemble Type	*Sen* (%)	*Spe* (%)	*Acc* (%)	*AUC*
ATP168	No Ensemble	77.6	77.9	77.9	0.859
	Maximum Ensemble	77.8	78.0	78.0	0.860
	Minimum Ensemble	78.3	78.6	78.6	0.868
	Mean Ensemble	78.4	78.8	78.8	0.867
	MAdaBoost Ensemble	79.1	79.8	79.8	0.873
ATP227	No Ensemble	80.1	80.2	80.2	0.882
	Maximum Ensemble	80.0	80.5	80.5	0.882
	Minimum Ensemble	80.8	81.1	81.1	0.889
	Mean Ensemble	80.9	81.1	81.1	0.889
	MAdaBoost Ensemble	81.5	82.0	82.0	0.895

4.3. Enhancing performance by SVM ensemble

Table 4 presents the performance comparison of different classifier ensemble strategies with *LogisticPSSM+PSS* feature on ATP168 and ATP227 with five-fold cross-validation tests, while Fig. 5 illustrates the ROC curves of single SVM (no ensemble) and *MAdaBoost* ensembled SVMs with *LogisticPSSM+PSS* feature on ATP227 over five-fold cross-validation. Note that in the presented results, the number of base SVM classifiers (L) is set to be 5.

It is found that although results from different ensemble strategies vary a little bit, they are all better than a single classifier model. Taking *MAdaBoost* ensemble as an example, the Accuracies (*Acc*) of the ensembled SVMs are 79.8% and 82.0% on ATP168 and ATP227, respectively, and about 2% improvements were achieved on both datasets. Other three evaluation indexes (*Sen*, *Spe*, and *AUC*) were also improved consistently. These results demonstrate that different sampled subsets can complement with each other and thus can yield better results with ensemble approach. At the same time, among the 4 ensemble strategies tested in this study, the MAdaBoost approach is found better than others as shown in Table 4.

4.4. Comparison with existing predictors

In this section, we will compare TargetATP with existing predictors for protein-ATP binding residues prediction. Note that in TargetATP, the combined features of *LogisticPSSM* and *PSS* are used as the input, and the *MAdaBoost* ensemble scheme was applied.

4.4.1. Comparison with ATPint

To the best of our knowledge, the ATPint [5] is the first predictor that was particularly designed for predicting protein-ATP binding residues from protein primary sequence. The ATPint was developed based on PSSM-based feature and the SVM was used to perform classification. In [5], Chauhan et al. tried several different thresholds, and the threshold, where sensitivity and specificity are nearly equal in order to make the balance between sensitivity and specificity, was chosen for the final reporting, the idea of which is quite similar to that of the *balanced evaluation* as we defined in Section 3.1.

Table 5 illustrates the comparison results between the TargetATP and the ATPint on ATP168 over five-fold cross-validation under *balanced evaluation*. Note that the threshold for balanced evaluation is identified to be 0.16, by which the value of the false positive rate ($FPR=FP/(FP+TN)$) is equal to that of the false negative rate ($FNR=FN/(FN+TP)$).

From Table 5, it is easy to find that the TargetATP outperforms the ATPint on all the five evaluation indexes. Note that in [5], the authors reported that the *MCC* is 0.50, and the corresponding *Sen*, *Spec*, and *Acc* are 74.4%, 75.8%, and 75.1%, respectively. However, according to the *Sen*, *Spec*, and *Acc* they reported, together with the numbers of positive and negative samples in ATP168, we can calculate that the *MCC* should be 0.249.

4.4.2. Comparison with ATPsite and NsitePred

The performances of the ATPsite and NsitePred were reported based on ATP227 with five-fold cross-validation [6,7]. To fairly compare the TargetATP with them, we also performed five-fold cross-validation on the same dataset. Table 6 illustrates the performance comparison of the ATPint, ATPsite, NsitePred, and TargetATP on ATP227.

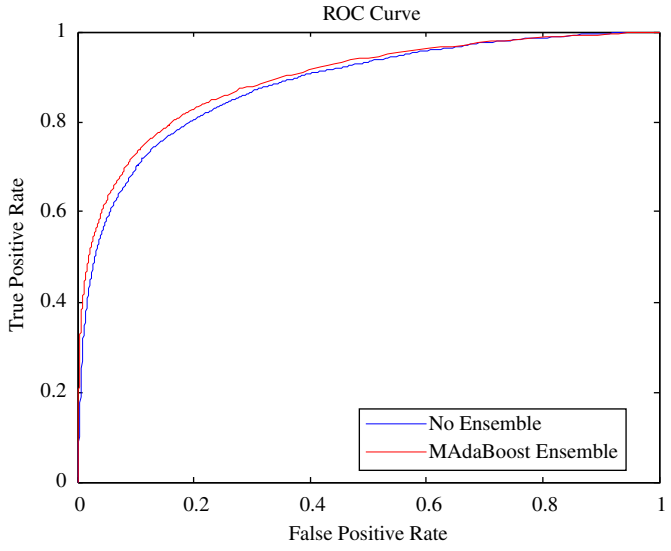


Fig. 5. ROC curves of no ensemble and *MAdaBoost* ensemble with *LogisticPSSM+PSS* feature on ATP227.

Table 5
Performance comparison of the TargetATP with ATPint on ATP168 over five-fold cross-validation with balanced evaluation.

Predictor	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*	*AUC*
ATPint[a]	74.4	75.8	75.1	0.249	0.823
TargetATP	79.1	79.8	79.8	0.308	0.873

[a] Data excerpted from [5].

We have reported the performance of the TargetATP on ATP227 over five-fold cross-validation with *balanced evaluation*, as shown in Table 4. However, ATPsite and NsitePred did not consider this matter and reported the results with *imbalanced evaluation*, which makes it impossible to directly compare the current model with ATPsite and NsitePred according to Table 4. In view of this, we also chose the threshold (0.78 for the TargetATP on ATP227 obtained by maximizing the *MCC*) for reporting prediction results in an unbalanced manner similar to that in ATPsite and NsitePred, which is illustrated in Table 6. From Table 6, we can find that the TargetATP significantly outperforms ATPint. In addition, the TargetATP also performs better than ATPsite on all the five considered evaluation indexes. The *AUC* and the *MCC* of TargetATP are 0.895 and 0.501, which are 4.1% and 6.8% higher than those of ATPsite, respectively. As to other three evaluation indexes, TargetATP also outperforms ATPsite. In addition, the performance of the TargetATP is also slightly outperforms the NsitePred, which is the most recently released protein-ATP binding residues predictor. We also evaluated the experimental results in Table 6 using a paired *t*-test [40]. If the resulting *p*-value is below the desired significance level (0.05 in this study), the performance difference between two methods is considered to be statistically significant. By this test, we found that the *MCC* and *AUC* of the TargetATP are statistically significantly better than that of all the listed predictors.

To further demonstrate the generalization ability of the TargetATP, an independent dataset from [7], which consists of 17 protein sequences that have not been used for training TargetATP, were used. Table 7 lists the performance comparison of different predictors on the independent dataset. From Table 7, it was clearly found that the TargetATP achieves the best performance on the independent dataset. The *AUC* and *MCC* of the TargetATP are 0.912 and 0.542, which are 3.7% and 6.6% higher that of the second-best performer (NsitePred), respectively. In addition, other three evaluation indexes, i.e., *Sen*, *Spe* and *Acc*, are also consistently better than that of other three listed predictors. This demonstrates that TargetATP has good generalization capability for ATP-binding residues prediction.

Table 6
Performance comparison of the TargetATP with three most recently released protein-ATP binding residues predictors on ATP227.

Predictor	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*		*AUC*	
				Value	*Sig.*	*Value*	*Sig.*
ATPint[a]	53.9	65.1	64.8	0.078	+	0.627	+
ATPsite[a]	36.1	98.8	96.2	0.433	+	0.854	+
NsitePred[a]	44.4	98.2	96.0	0.460	+	0.861	+
TargetATP	41.2	99.0	96.6	0.501		0.895	

[a] Data excerpted from [7]. The significance of the differences between TargetATP and the other predictors are measured for the *MCC* and *AUC* and they are given in the '*Sig.*' columns. The '+' means that the TargetATP is statistically significantly better with p-value < 0.05.

Table 7
Performance comparison of the TargetATP with three most recently released protein-ATP binding residues predictors on independent dataset.

Predictor	*Sen* (%)	*Spe* (%)	*Acc* (%)	*MCC*	*AUC*
ATPint[a]	51.2	66.0	65.5	0.066	0.606
ATPsite[a]	36.7	99.1	96.9	0.451	0.868
NsitePred[a]	46.0	98.5	96.7	0.476	0.875
TargetATP	48.9	98.9	96.9	0.542	0.912

[a] Data excerpted from [7].

4.4.3. Comparison with Firoz et al.'s method

Lastly, we demonstrate the efficacy of the proposed TargetATP on dataset ATP131. Table 8 illustrates the performance comparison between the TargetATP and the Firoz et al.'s method [14] on ATP131.

From Table 8, it was found that TargetATP again achieves satisfactory results comparing with the Firoz et al.'s method on ATP131. Although the *Sen* of the TargetATP were inferior to that of the Firoz et al.'s method, the other four measures are better.

4.5. Case study

In this section, two protein sequences (PDB ID: 1DV2_A and 2C8V_A), which have not been used for training our TargetATP, were used for case studies.

The prediction results for 1DV2_A and 2C8V_A with ATPint were obtained by feeding the two sequences to the web server that is available at http://www.imtech.res.in/raghava/atpint/. The ATPint provides a user-defined threshold to obtain the prediction results and the default threshold is 0.2. However, we found that prediction results for the two chosen protein sequences under the default threshold (0.2) were quite poor. Thus, we tried several different thresholds and the best ones (0.60 for 1DV2_A and 0.70 for 2C8V_A) were then chosen for reporting. As ATPsite does not provide a web server, thus it is not included in the case study.

The prediction results for 1DV2_A and 2C8V_A with NsitePred were obtained by feeding the two sequences to the web server that is available at http://biomine.ece.ualberta.ca/nSITEpred/.

Figs. 6 and 7 illustrate the prediction results of different predictors for 1DV2_A and 2C8V_A, respectively, where the green stars denote the observed (true) binding residues, red stars denote the binding residues predicted by TargetATP, the blue stars mean the binding residues predicted by ATPint, and the black stars mean the binding residues predicted by NsitePred. The probability curves obtained by TargetATP, ATPint, and NsitePred were also plotted with red, blue and black colors, respectively.

From Figs. 6 and 7, it is easy to find that the TargetATP and NsitePred significantly outperform the ATPint. In addition, TargetATP outperforms NsitePred for 1DV2_A, and is comparable to NsitePred for 2C8V_A. Taking 1DV2_A, the TargetATP clearly outperforms the NsitePred. The TargetATP correctly predicted 14 out of the 16 binding residues and only 3 non-binding residues are mistakenly identified as binding residues (3 false positives). In contrast, the NsitePred correctly identified only 10 out of the 16 binding residues while with 17 false positives. As for 2C8V_A, the performance of TargetATP is comparable to that of the NsitePred. The NsitePred correctly identified 16 out of the 19 binding residues with 3 false positives; while the TargetATP correctly predicted 15 binding residues also with 3 false positives.

Table 8
Performance comparison of between the TargetATP and Firoz et al.'s method on ATP131.

Predictor	Sen (%)	Spe (%)	Acc (%)	F-measure	AUC
Firoz et al.'s method	63.4	91.8	90.2	0.410	0.827
TargetATP	58.6	93.1	91.2	0.418	0.841

5. Conclusions and future directions

In this study, we have developed a new protein-ATP binding residues predictor with high accuracy, by integrating multi-view features and applying classifier ensemble. The good performances of the current model come from the systematic analysis and use of the most discriminative features and SVM ensemble constructed based on multiple under-sampled datasets. As many of the important bioinformatics topics of identifying the functional residues in protein sequences can be formulated as the imbalanced learning problems, current study provides an effective solution that can be applied.

Our future work includes the following two directions. (1) Further improving the prediction performance of the TargetATP by incorporating new developed feature extraction methods and powerful classifiers. For example, L_1-reguralized logistic regression-based feature selection method [41] has been successfully applied to active site prediction; sparse inverse covariance estimation method has been successfully used for structural contact prediction on large multiple sequence alignments [42]. These new emerged methods provide promising routes for us to further improve the performance of the TargetATP. (2) Besides ATP studied in this study, there are many other ligands that can bind protein targets, it is also important to investigate how to effectively discriminate the different types of ligands to reveal the various molecular binding mechanisms.

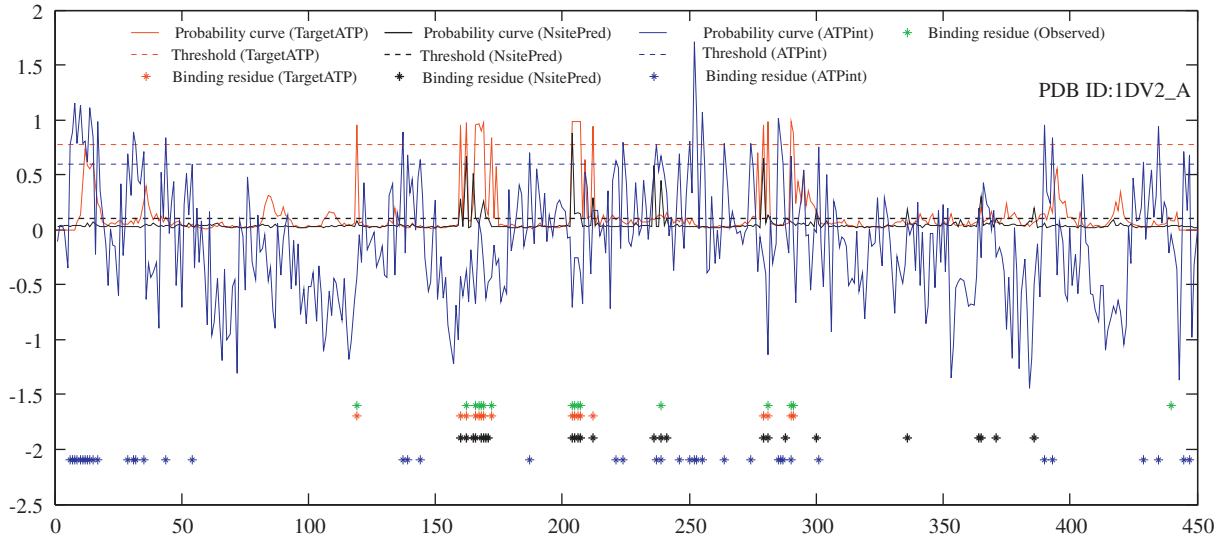


Fig. 6. Binding residues predicted by TargetATP, ATPint, and NsitePred for the protein 1DV2_A. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

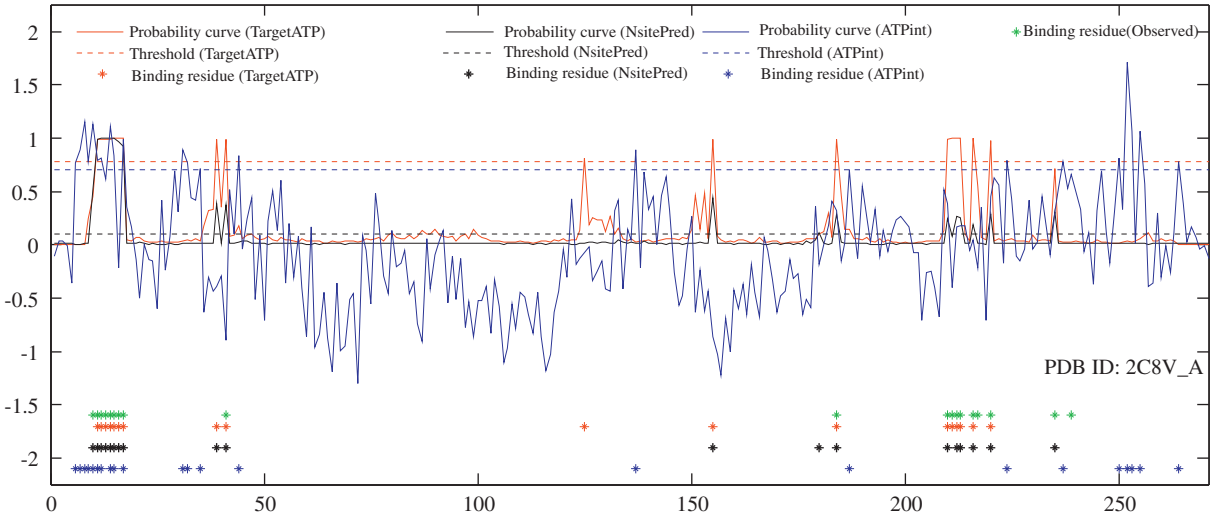


Fig. 7. Binding residues predicted by TargetATP, ATPint, and NsitePred for protein 2C8V_A. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

Acknowledgment

The authors wish to thank the three anonymous reviewers for valuable suggestions and comments, which are very helpful for improvement of this paper. This work was supported by the National Natural Science Foundation of China (No. 91130033, 61175024, 61233011, and 61222306), the Natural Science Foundation of Jiangsu (No. BK2011371), Jiangsu Postdoctoral Science Foundation (No. 1201027C), A Foundation for the Author of National Excellent Doctoral Dissertation of PR China (No. 201048), the National Science Fund for Distinguished Young Scholars (No.61125305), Industry-Academia Cooperation Innovation Fund Projects of Jiangsu Province (No. BY2012022), Shanghai Science and Technology Commission (No.11JC1404800).

Appendix A. Proof of balanced threshold selection

By definition, we have

$$FNR = FN/(FN+TP) \tag{A.1}$$

$$FPR = FP/(FP+TN) \tag{A.2}$$

$$TPR = TP/(FN+TP) \tag{A.3}$$

Clearly,

$$FNR+TPR = FN/(FN+TP)+TP/(FN+TP) = 1 \tag{A.4}$$

For the black point in Fig. 4, we have

$$FPR+TPR = 1 \tag{A.5}$$

Subtracting Eq. (A.4) from Eq. (A.5), we have

$$FPR-FNR = 0 \tag{A.6}$$

Thus,

$$FPR = FNR \tag{A.7}$$

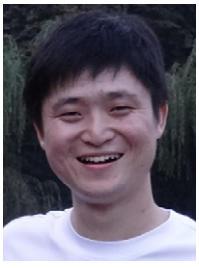
From Eq. (A.7), we can find that the black ROC point in Fig. 4 corresponds to the threshold, by which the value of the *false positive rate* (*FPR*) is equal to that of the *false negative rate* (*FNR*), for balanced evaluation.

References

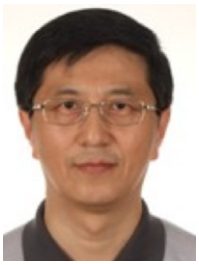
[1] N.A. Campbell, B. Williamson, R.J. Heyden, Biology: Exploring Life, Pearson Prentice Hall, Boston, Massachusetts, 2006.
[2] A. Maxwell, D.M. Lawson, The ATP-binding site of type II topoisomerases as a target for antibacterial drugs, Curr. Top. Med. Chem. 3 (3) (2003) 283–303.
[3] A. Abbaspour, L. Baramakeh, Application of principle component analysis-artificial neural network for simultaneous determination of zirconium and hafnium in real samples, Spectrochim. Acta Part A – Mol. Biomol. Spectrosc. 64 (2) (2006) 477–482.
[4] I. Nobeli, R.A. Laskowski, W.S.J. Valdar, J.M. Thornton, On the molecular discrimination between adenine and guanine by proteins, Nucleic Acids Res. 29 (21) (2001) 4294–4309.
[5] J.S. Chauhan, N.K. Mishra, G.P. Raghava, Identification of ATP binding residues of a protein from its primary sequence, BMC Bioinformatics 10 (2009) 434.
[6] K. Chen, M.J. Mizianty, L. Kurgan, ATPsite: sequence-based prediction of ATP-binding residues, Proteome. Sci. 9 (Suppl 1) (2011) S4.
[7] K. Chen, M.J. Mizianty, L. Kurgan, Prediction and analysis of nucleotide-binding residues using sequence and sequence-derived structural descriptors, Bioinformatics 28 (3) (2012) 331–341.
[8] Z.H. Zhou, X.Y. Liu, On multi-class cost-sensitive learning, Comput. Intell. 26 (3) (2010) 232–257.
[9] 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.
[10] H. Haibo, E.A. Garcia, Learning from Imbalanced Data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[11] Z.Y. Lin, Z.F. Hao, X.W. Yang, X.L. Liu, in: R. Huang (Ed.), Several SVM Ensemble Methods Integrated with Under-Sampling for Imbalanced Data Learning, ADMA, Springer-Verlag, Berlin Heidelberg, 2009, pp. 536–554.
[12] D. Mease, A.J. Wyner, A. Buja, Boosted classification trees and class probability/quantile estimation, J.Mach. Learn. Res. 8 (2007) 409–439.
[13] X.Y. Liu, J.X. Wu, Z.H. Zhou, Exploratory undersampling for class-imbalance learning, IEEE Trans. Syst. Man Cybern. Part B –Cybern. 39 (2) (2009) 539–550.
[14] A. Firoz, A. Malik, K.H. Joplin, Z. Ahmad, V. Jha, S. Ahmad, Residue propensities, discrimination and binding site prediction of adenine and guanine phosphates, BMC Biochem. 12 (2011) 20.
[15] J.C. Jeong, X. Lin, X.W. Chen, On position-specific scoring matrix for protein function prediction, IEEE/ACM Trans. Comput. Biol. Bioinformatics 8 (2) (2011) 308–315.
[16] Y.N. Zhang, D.J. Yu, S.S. Li, Y.X. Fan, Y. Huang, H.B. Shen, Predicting protein-ATP binding sites from primary sequence through fusing bi-profile sampling of multi-view features, BMC Bioinformatics 13 (2012) 118.
[17] D.J. Yu, H.B. Shen, J.Y. Yang, SOMPNN: an efficient non-parametric model for predicting transmembrane helices, Amino Acids 42 (6) (2012) 2195–2205.
[18] M.H. Zangooei, S. Jalili, Protein secondary structure prediction using DWKF based on SVR-NSGAII, Neurocomputing 94 (1) (2012) 87–101.
[19] A. Pierleoni, P.L. Martelli, R. Casadio, MemLoci: predicting subcellular localization of membrane proteins in eukaryotes, Bioinformatics 27 (9) (2011) 1224–1230.
[20] H.B. Shen, K.C. Chou, A top-down approach to enhance the power of predicting human protein subcellular localization: Hum-mPLoc 2.0, An. Biochem. 394 (2) (2009) 269–274.
[21] A.A. Schaffer, Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements, Nucleic Acids Res. 29 (2001) 2994–3005.
[22] D.T. Jones, Protein secondary structure prediction based on position-specific scoring matrices, J. Mol. Biol. 292 (2) (1999) 195–202.
[23] V.N. Vapnik, Statistical Learning Theory, Wiley-Interscience, New York, 1998.
[24] R.E. Fan, P.H. Chen, C.J. Lin, Working set selection using second order information for training SVM, J. Mach. Learn. Res. 6 (2005) 1889–1918.

[25] L. Xu, S. Amari, Combining classifiers and learning mixture-of-experts, in: Juan Ramón Rabuñal Dopico, Julian Dorado, Alejandro Pazos (Eds.), Encyclopedia of Artificial Intelligence, IGI Global (IGI) Publishing Company, 2009, pp. 218–326.
[26] L. Nanni, A novel ensemble of classifiers for protein fold recognition, Neurocomputing 69 (16–18) (2006) 2434–2437.
[27] L. Nanni, Ensemble of classifiers for protein fold recognition, Neurocomputing 69 (7–9) (2006) 850–853.
[28] J. Wu, M.L. Li, L.Z. Yu, C. Wang, An ensemble classifier of support vector machines used to predict protein structural classes by fusing auto covariance and pseudo-amino acid composition, Protein J. 29 (1) (2010) 62–67.
[29] J. Kittler, F.M. Alkoot, Sum versus vote fusion in multiple classifier systems, IEEE Trans. Pattern Anal. Mach. Intell. 25 (1) (2003) 110–115.
[30] L.A. Alexandre, A.C. Campilho, M. Kamel, On combining classifiers using sum and product rules, Pattern Recognition Lett. 22 (12) (2001) 1283–1289.
[31] T.K. Kim, J. Kittler, Combining classifier for face identification at unknown views with a single model image, in: Proceedings of the Structural, Syntactic, and Statistical Pattern Recognition, vol. 3138, 2004, pp. 565–573.
[32] D.M.J. Tax, M. van Breukelen, R.P.W. Duin, J. Kittler, Combining multiple classifiers by averaging or by multiplying? Pattern Recognition 33 (9) (2000) 1475–1485.
[33] J. Grim, J. Kittler, P. Pudil, P. Somol, Combining multiple classifiers in probabilistic neural networks, Multiple Classifier Syst. 1857 (2000) 157–166.
[34] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, John Wiley & Sons, 2004.
[35] Y. Freund, R.E. Schapire, A decision–theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[36] H. Haibo, E.A. Garcia, Learning from Imbalanced Data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[37] T. Jo, N. Japkowicz, Class imbalances versus small disjuncts, ACM SIGKDD Explorations Newslett. 6 (1) (2004) 40–49.
[38] D.J. Yu, H.B. Shen, J.Y. Yang, SOMRuler: a novel interpretable transmembrane helices predictor, IEEE Trans. Nanobiosci. 10 (2) (2011) 121–119.
[39] H. Shen, J.J. Chou, MemBrain: improving the accuracy of predicting transmembrane helices, PLoS One 3 (6) (2008) e2399.
[40] J. Yang, L. Zhang, J.Y. Yang, D. Zhang, From classifiers to discriminators: a nearest neighbor rule induced discriminant analysis, Pattern Recognition 44 (7) (2011) 1387–1402.
[41] S. Sankararaman, F. Sha, J.F. Kirsch, M.I. Jordan, K. Sjolander, Active site prediction using evolutionary and structural information, Bioinformatics 26 (5) (2010) 617–624.
[42] D.T. Jones, D.W. Buchan, D. Cozzetto, M. Pontil, PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments, Bioinformatics 28 (2) (2012) 184–190.

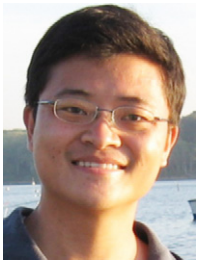
Dong-Jun Yu received the B.S. degree in computer science and the MS degree in artificial intelligence from Jiangsu University of Science and Technology in 1997 and 2000, respectively, and the Ph.D. degree in pattern analysis and machine intelligence from Nanjing University of Science and Technology in 2003. In 2008, he acted as an academic visitor at University of York in UK. He is currently an associate professor in the School of Computer Science and Engineering of Nanjing University of Science and Technology. His current interests include pattern recognition, data mining and bioinformatics.



Jun Hu received his B.S. degree in computer science from Anhui Normal University, China in 2011. Currently, he is working towards the M.S. degree in the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. His research interests include bioinformatics, data mining, and pattern recognition.



Zhen-Min Tang received the B.S. degree and M.S. degree in Computer Science from Nanjing University of Science and Technology (NUST), Nanjing, China. He is now a Professor in the School of Computer Science and Engineering at NUST. He is the author of over 40 scientific papers in pattern recognition, image processing, and artificial intelligence. His current interests are in the areas of pattern recognition, image processing, artificial intelligence, and expert system.



Hong-Bin Shen received his Ph.D. degree from Shanghai Jiaotong University China in 2007. He was a postdoctoral research fellow of Harvard Medical School from 2007 to 2008. Currently, he is a professor of Institute of Image Processing and Pattern Recognition, Shanghai Jiaotong University. His research interests include data mining, pattern recognition, and bioinformatics. Dr. Shen has published more than 60 papers and constructed 20 bioinformatics severs in these areas and he serves the editorial members of several international journals.

Jian Yang received the B.S. degree in mathematics from the Xuzhou Normal University in 1995. He received the MS degree in applied mathematics from the Changsha Railway University in 1998 and the PhD degree from the Nanjing University of Science and Technology (NUST), on the subject of pattern recognition and intelligence systems in 2002. In 2003, he was a postdoctoral researcher at the University of Zaragoza. From 2004 to 2006, he was a Postdoctoral Fellow at Biometrics Centre of Hong Kong Polytechnic University. From 2006 to 2007, he was a Postdoctoral Fellow at Department of Computer Science of New Jersey Institute of Technology. Now, he is a professor in the School of Computer Science and Engineering of NUST. He is the author of more than 80 scientific papers in pattern recognition and computer vision. His journal papers have been cited more than 1600 times in the ISI Web of Science, and 2800 times in the Web of Scholar Google. His research interests include pattern recognition, computer vision and machine learning. Currently, he is an associate editor of Pattern Recognition Letters and IEEE Trans. Neural Networks, respectively.

Jing-Yu Yang received the B.S. Degree in Computer Science from NUST, Nanjing, China. From 1982 to 1984 he was a visiting scientist at the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign. From 1993 to 1994 he was a visiting professor at the Department of Computer Science, Missouri University. And in 1998, he acted as a visiting professor at Concordia University in Canada. He is currently a professor and Chairman in the department of Computer Science at NUST. He is the author of over 150 scientific papers in computer vision, pattern recognition, and artificial intelligence. He has won more than 20 provincial awards and national awards. His current research interests are in the areas of pattern recognition, robot vision, image processing, data fusion, and artificial intelligence.