PETs: A Stable and Accurate Predictor of Protein-Protein Interacting Sites Based on Extremely-Randomized Trees

Bin Xia, Hong Zhang, Qianmu Li*, and Tao Li

***Abstract*—Protein-protein interaction (PPI) plays crucial roles in the performance of various biological processes. A variety of methods are dedicated to identify whether proteins have interaction residues, but it is often more crucial to recognize each amino acid. In practical applications, the stability of a prediction model is as important as its accuracy. However, random sampling, which is widely used in previous prediction models, often brings large difference between each training model. In this paper, a Predictor of protein-protein interaction sites based on Extremely-randomized Trees (PETs) is proposed to improve the prediction accuracy while maintaining the prediction stability. In PETs, a cluster-based sampling strategy is proposed to ensure the model stability: first, the training dataset is divided into subsets using specific features; second, the subsets are clustered using K-means; and finally the samples are selected from each cluster. Using the proposed sampling strategy, samples which have different types of significant features could be selected independently from different clusters. The evaluation shows that PETs is able to achieve better accuracy while maintaining a good stability. The source code and toolkit are available at https://github.com/BinXia/PETs.**

***Index Terms*—ETs, PETs, sampling strategy, stability.**

I. Introduction

PROTEIN-PROTEIN interactions (PPIs) play crucial roles in the performance of various biological processes. The protein which is responsible for cellular mediation could be easily obtained in the sequence-based fully genomes; however, the functionality of protein and the mediation of cellular behaviors are still ambiguous. PPIs are used to realize the mediation of many cellular processes, and the mediation is directly associated with dominating almost all biochemical reactions in the living cells [1], [2]. Therefore, in order to better understand various cellular mediations, mechanisms of cellular processes and the development of diseases, PPIs should be fully explored. However, the biological-experiment-based method used to search for PPIs is often complex and time-consuming even for drosophila melanogaster [3].

Facing such opportunities and challenges, many efficient and effective methods of PPI site prediction have been proposed [4]–[9]. For example, Shen *et al.* [10] predicted PPI networks based on sequence information, You *et al.* [11] presented ensemble extreme learning machines to predict PPIs between protein pairs, Šikić *et al.* [12], and Piero *et al.* [13] constructed a prediction model using 3D structure information. These prediction methods are excellent in recognizing an interacting region of a protein. However, the prediction of the interacting region is easier than the recognition of each residue of a protein. The residue prediction problem often has imbalanced class distribution because the interacting residues are much less than the non-interacting ones.

For residue-scale interacting prediction, many predictors such as SPPIDER [14], ISIS [15], PSIVER [16], LORIS [17] have been proposed using different classifiers (support vector machines (SVMs), neural networks (NNs), naive Bayes (NB), and L_1-regularized logistic regression (L_1-logreg) [18]) and features. However, they did not deal with the imbalance classification very well as PSIVER [16], ISIS [15], and SPPIDER [14] have a huge difference between Recall and Specificity which will be mentioned in Section II-F. In addition, we found that the definition of interacting residues based on complex formation is proposed in 1997 [19], [20], and the definition of surface using NACCESS [21] is based on the algorithm of Lee and Richards in 1971 [22]. Gallet *et al.* [23] and Bock *et al.* [24] also had their own definitions of PPI. There are also some other authoritative definitions of interface residues. As a result, a flexible and robust prediction model that is able to fit different definitions is needed.

Besides the class imbalance problem, another problem in residue-scale interaction predictions is the stability of random sampling used in previous methods. Note that labels are totally unknown in practical applications [25]. Although random sampling is quite effective as the basis of most sampling strategies, its stability can not be guaranteed since each step of random

Manuscript received February 19, 2015; revised August 20, 2015; accepted October 06, 2015. Date of publication October 27, 2015; date of current version January 07, 2016. The work is partially supported by the Natural Science Foundation of China under Grant No. 61272419, the Prospective Future Network Research Projects of Jiangsu Province under Grant No. BY2013095-3-02, the Prospective Studies and Research Projects of Jiangsu Province under Grant No. BY2013039, Grant No. BY2013037, and Grant No. BY2014089, and the Jiangsu Key Laboratory of Image and Video Understanding for Social Safety (Nanjing University of Science and Technology), Grant No. 30920140122007. The work of T. Li is partially supported by U.S. National Science Foundation under Grant DBI-0850203. *Asterisk indicates corresponding author.*

B. Xia and H. Zhang are with School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, P. R. China (e-mail: ben.binxia@gmail.com).

*Q. Li is with School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China, and also with Jiangsu Collaborative Innovation Center of Social Safety Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: liqianmu@126.com).

T. Li is with School of Computing and Information Sciences, Florida International University, Miami, FL, 33199 USA; and also with School of Computer Science & Technology, Nanjing University of Posts and Telecommunications (NJUPT), Nanjing 210046, China (e-mail: taoli@cs.fiu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNB.2015.2491303

sampling selects an instance randomly without considering the relationships among instances [26], [27].

In this paper, we propose a **P**redictor of Protein-Protein interaction sites based on **E**xtremely-randomized **Tree**s (PETs) where a new sampling approach based on K-means clustering is employed. Extremely-randomized trees (ETs) is used as the classifier using sequence-based feature position specific scoring matrix (PSSM), protein second structure (PSS), predicted solvent accessibility (PSA), and predicted relative solvent accessibility (PRSA). ETs is proved as an excellent classifier in many conditions, and it constructs an ensemble of decision trees in a top-down manner.

The rest of this paper is organized as follows. In Section II, we introduce our datasets and details of PETs. In Section III, we present the experimental results. Finally, we summarize and conclude this work in Section IV.

II. Materials and Methods

A. Benchmark Dataset

The datasets used in this paper is divided into the training dataset and the test dataset. Dset186 mentioned in PSIVER [16] is used as our training dataset. Dset186, which contains 186 heterodimeric, non-transmembrane and transient protein sequences, was extracted from Protein Data Bank (PDB) [28], and the structures of the protein sequences in Dset186 was solved through X-ray Crystallography with resolution less than or equal to 3.0 Å. Dtestset72 and PDBset164 are both used as the test datasets. Dtestset72 mentioned in PSIVER has 72 non-overlapping protein sequences [16]. For comparing with PSIVER and LORIS, PDBset164 which is proposed in SPRINGS is also used as a test dataset [29]. In [30], Mihel *et al.* proposed Protein Structure and Interaction Analyzer (PSAIA) which ensembles 4 kinds of meaningful interacting definition. Each definition is accepted to redefine Dset186, Dtestset72, and PDBset164, and their detailed descriptions are listed below. The distribution of interacting and non-interacting residues is displayed in Appendix A.

1) Dset-ASAChange and Dset-Murakami: This definition mainly uses solvent accessibility (SA) of each amino acid. The residue is marked as a surface amino acid with the relative solvent accessibility (rSA) $<5\%$ [21], [22], and a surface residue is considered as an interface if its lost absolute solvent accessibility (SA) <1.0 Å^2 [19], [20]. That is the definition Murakami *et al.* [16] used. Notably, although PSAIA provides the program to define the Dset-ASAChange, labels from Murakami *et al.* [16] are quite different from the labels that PSAIA outputs in Dtestset72.

2) Dset-ANDistance: The aspect of Atom Nucleus Distance (ANDistance) was introduced by Ofran *et al.* [31]. A residue is considered as an interface if any of its atoms has the specified distance away from the atoms of residue in the opposite chain. The threshold of distance is always $\leq$6 Å. 4.50 Å is the criterion used in this paper.

3) Dset-AVWRDistance: Atom Van der Waals Radii Distance(AVWRDistance) was the definition proposed by Aytuna *et al.* [32]. The definition is almost the same as the ANDistance, but only the distance is changed to Atom Van der Waals Radii Distance. 0.5 Å is treated as the reference threshold in our work.

4) Dset-PIADA: Mihel *et al.* [30] proposed Protein Interaction Atom Distance Algorithm (PIADA) including 4 kinds of interaction: ionic, polar, Van der Waals, and hydrophobic. The ionic interface is defined as the distance between ionic atoms $<$6 Å. The polar interface is $<$4.7 Å between polar atoms. Hydrophobic interface is marked as interaction that the distance between any two amino acid is $<$4.7 Å when amino acids are non-polar in the following: Ala, Ile, Leu, Met, Phe, Pro, Val. Van der Waals interface is defined using the following equation:

$$D_{ij} < r_i + r_j + 1.125, \tag{1}$$

where D_{ij} denotes the distance between residue i and j, r_i, r_j represent Van der Waals radii of residue i and j, the unit of the equation is Å.

Notably, there is an extra residue CYS at 56th of 2J3R_A when we run PSAIA. For the consistency with previous works, the residue CYS is removed because it does not affect the labels nearby. For the integrity of the features, the first and last 4 residues of each protein are removed (because a 9-residue sliding windows is used in our approach, more details can be found in Section II). If a special statement is not given, the following research samples are without the first and last 4 amino acids in each protein.

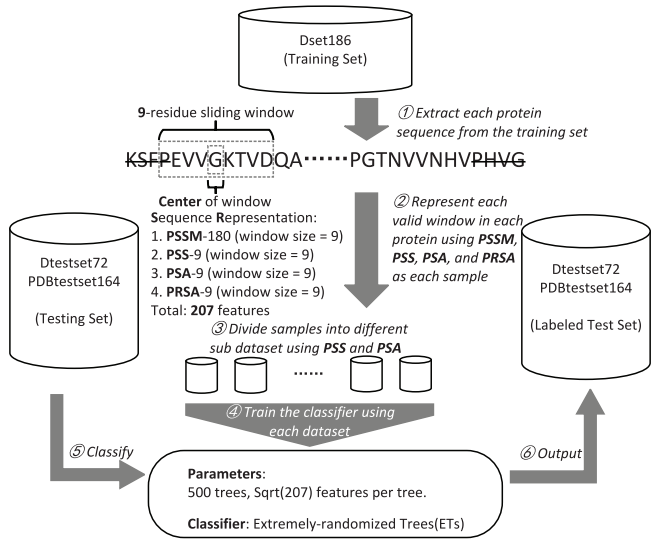


Fig. 1. The schematic diagram which describes the algorithm in our approach.

B. Schematic Diagram

A schematic diagram which describes our proposed approach is shown in Fig. 1. The approach consists of the following 3 steps: Step 1: The training set is first divided into several subsets; Step 2: Extremely-randomized trees (ETs) is then trained using samples selected from subsets clustered by K-means; and Step 3: ETs will output labels of residues in the test set.

C. Features Extraction and Sequence Representation

In proteomics, the protein is diverse and mysterious. Therefore, the extraction of features is difficult. Based on previous experiments [33], [34], the following features are used:

Feature I: *Position Specific Scoring Matrix (PSSM)*. PSSM includes the considerable evolutionary information of proteins although its generation is quite time-consuming using PSI-BLAST [35]. In order to compare with previous works, in this paper, BLAST+ is used with the same options (psiblast query protein num_iterations 3 db nr inclusion_ethresh 0.001

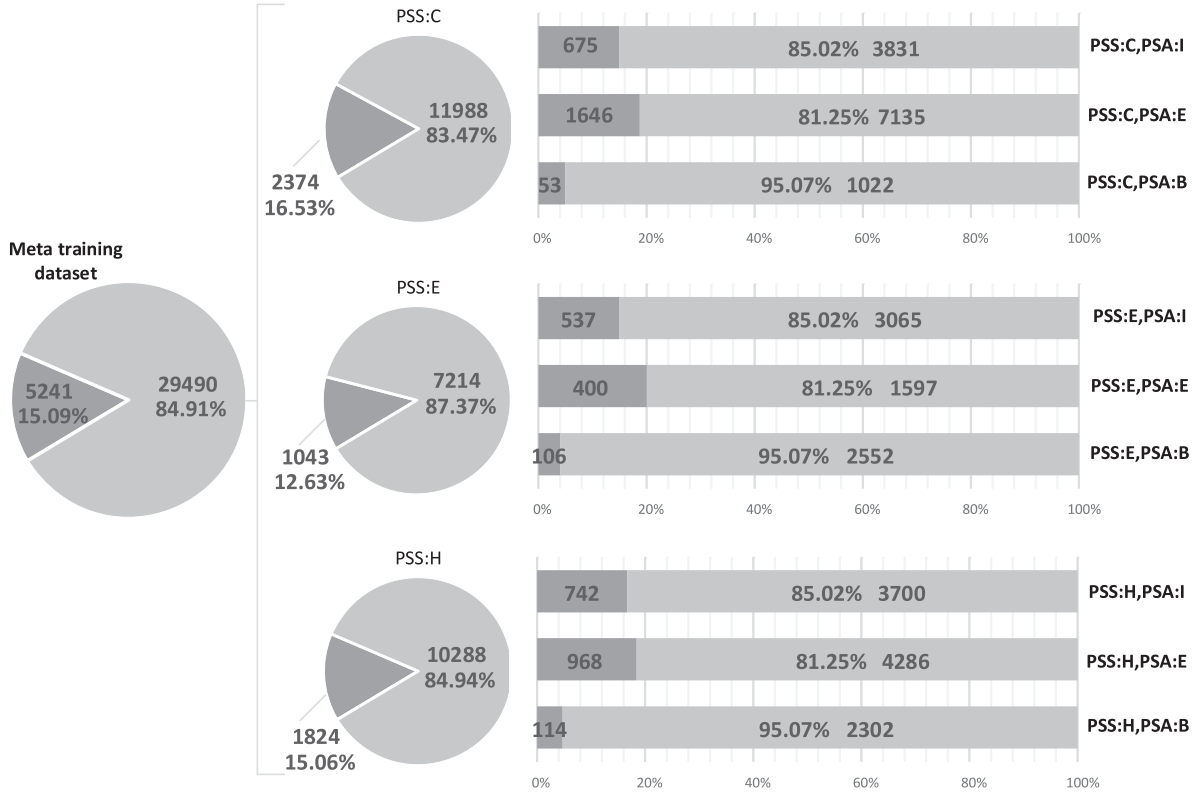


Fig. 2. The detail of dataset segmentation. The distribution of figure is the original definition of Dset186. The dark gray means the positive samples, the other color represents the negative ones. The pie on the left is the raw training dataset, which is displaying the distribution of Dset186, 3 pies in layer 1 (on the middle) are the segmentation of Dset186 based on classes (α-helix (H), β-sheet (E) and random coil (C)) of Protein Secondary Structure (PSS). The bars means the segmentation of each dataset in layer 1 based on classes (buried (B), intermediate (I), and exposed (E)) of Predicted Solvent Accessibility(PSA).

out_ascii_pssm pssm) in the research of K. Dhole *et al.* against the NCBI non-redundant protein sequence database [17], [36]. BLAST+ is available at ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/, and NCBI non-redundant protein sequence database and its blast database are available at ftp://ftp.ncbi.nlm.nih.gov/blast/db/. Many previous experiments demonstrated that a nine-residue sliding window was the best choice in the research of protein-protein interacting prediction [12], [14]–[17]. In this paper, a nine-residue sliding window is also used to extract features from PSSM (the number of features: $180 = 20 * 9$), and the feature scores are not normalized. PSSM can be used as the standard to normalize other features. The first and last 4 residues of every protein sequence, which lack the information in a nine-residue sliding window, are not involved in our research.

Feature II: *Protein Secondary Structure (PSS)*. PSS is the conformation of repetition in polypeptide chains with rules, and the common PSS has 3 types of construction: α-helix (H), β-sheet (E), and random coil (C). In the previous research of targeting protein-ligand binding sites prediction, PSS had played a very crucial role [37]. In this paper, PSS is extracted as a nine-residue-sliding-window feature (the number of feature: 9). SSpro, which is based on the sequence homology and the secondary structure of homologous protein, is employed to predict PSS. PSS is normalized using the ranges of PSSM, and the variable "pss2fea" is set to be the results in our python program (pss2fea = `C' : 0, `E' : 13, `H' : -13) [38]. SSpro is available at http://scratch.proteomics.ics.uci.edu/.

Feature III: *Predicted Solvent Accessibility (PSA)*. PSA information is predicted using SANN, which gives a three-state classification of solvent accessibility including buried (B), intermediate (I), and exposed (E). SANN is available at http://lee.kias.re.kr/newton/sann/ [39]. A variable "psa2fea" is also set to be the results in our python program

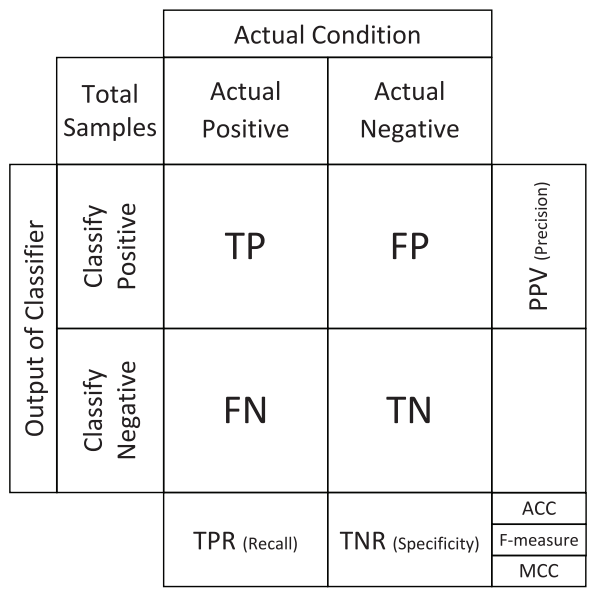


Fig. 3. The confusion matrix and relevant evaluation index. True Positive (TP): The number of residues classified as interacting correctly, False Positive (FP): The number of residues classified as interacting correctly incorrectly, False Negative (FN): The number of residues classified as non-interacting incorrectly, True Negative (TN): The number of residues classified as non-interacting correctly.

TABLE I
THE PERFORMANCE OF EACH FEATURE ON DSET186 USING LOOCV[1]

	AUC	Recall(%)	Specificity(%)	Precision(%)	Accuracy(%)	**MCC(%)**	F-measure(%)
PSSM[9]	0.649	58.8	61.6	26.9	60.1	16.0	33.9
	0.647	59.2	58.5	20.2	58.6	12.7	30.1
PSSM[9]+PSS[1]	0.651	59.4	61.4	27.2	60.2	16.5	34.3
	0.651	59.9	58.4	20.4	58.6	13.2	30.4
PSSM[9]+PSS[9]	0.655	60.5	61.0	27.2	60.0	16.9	34.4
	0.654	60.9	57.4	20.3	57.9	13.1	30.4
PSSM[9]+PSS[9]+PSA[1]	0.688	67.5	58.8	28.3	59.9	20.5	37.3
	0.685	57.4	56.0	21.4	57.7	16.8	32.5
PSSM[9]+PSS[9]+PSA[9]	0.691	67.9	59.1	28.7	60.3	21.1	37.8
	0.692	67.8	56.3	21.6	58.0	17.3	33.8
PSSM[9]+PSS[9]+PSA[1]+PRSA[1]	0.707	67.9	61.0	29.8	61.6	22.8	38.5
	0.706	68.4	57.6	22.3	59.2	18.7	33.6
PSSM[9]+PSS[9]+PSA[9]+PRSA[1]	0.714	68.8	61.5	30.5	62.4	24.0	39.5
	0.711	68.7	58.1	22.6	59.7	19.3	34.0
PSSM[9]+PSS[9]+PSA[1]+PRSA[9]	0.732	68.1	64.8	32.5	64.4	26.2	40.4
	0.730	68.5	59.9	23.3	61.2	20.5	34.8
PSSM[9]+PSS[9]+PSA[9]+PRSA[9]	0.735	68.3	65.7	33.3	65.4	**27.4**	41.4
	0.737	69.6	60.4	23.8	61.8	21.7	35.5
LORIS_total(our features)	-	65.4	67.8	33.1	66.9	26.8	41.4
	-	65.6	64.3	24.8	64.5	21.9	36.0
LORIS_valid(our features)	-	65.6	68.6	34.5	67.4	**28.0**	41.8
	-	66.4	64.3	24.8	64.6	22.3	36.1
LORIS	-	69.8	58.6	28.7	60.2	22.1	38.4
PSIVER	-	41.6	74.3	30.6	67.3	15.1	35.3

[1] Two different results for each feature combination are presented. The first line is obtained by averaging the experiment results of LOOCV. The second line is calculated using the accumulated confusion matrix.

TABLE II
THE PERFORMANCE (AVERAGE) OF WHOLE FEATURES USING LOOCV

Dataset	Features	Predictor	**MCC(%)**	F-measure(%)
Dset186	PETs	ETs	27.4	41.4
		L_1-logreg	**28.0**	41.8
	LORIS	ETs	19.6	36.3
		L_1-logreg	**22.1**	38.4
PDBtestset164	PETs	ETs	**19.4**	39.8
		L_1-logreg	17.2	38.5
	LORIS	ETs	**12.7**	36.6
		L_1-logreg	10.1	35.3

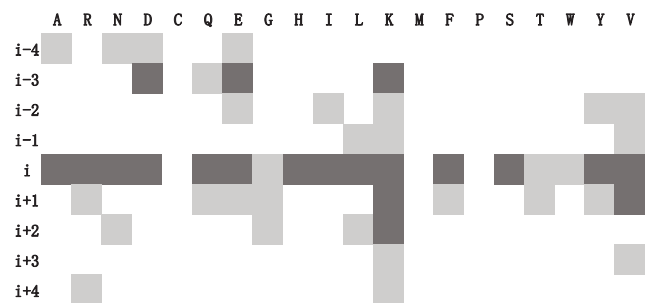


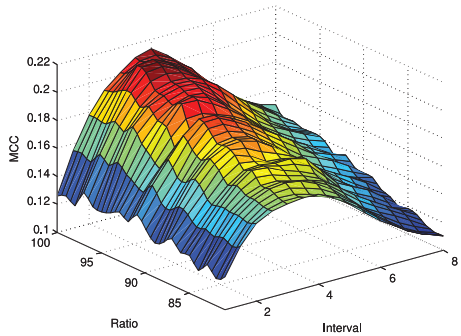
Fig. 4. The weight of PSSM based on a 9-residue sliding window in ETs. Top 20 are displayed in dark gray, and light gray shows other features in top 50 of PSSM.

$(\text{psa2num} = \text{`I'} : 0, \text{`E'} : 13, \text{`B'} : \text{-}13)$, and PSS is extracted as a nine-residue-sliding-window feature (the number of feature: 9). Notably, 1n2c_ABCD with 2000 residues is removed from our dataset, because SANN only accepts the protein with residues less than 1000, and it does not have an off-line version.

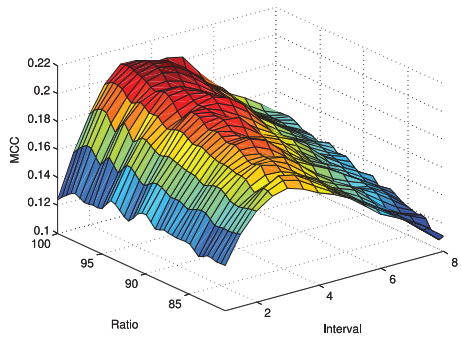
Feature IV: *Predicted Relative Solvent Accessibility (PRSA)*. PRSA, in this paper, is not same as PRSA LORIS used in [17]. Here PRSA is extracted using ACCpro20, which predicts the RSA using thresholds from 0% to 95% with 5% a step. This feature is first enlarged 20 times, then subtracted by 10 to have a range of $[-10, 10]$. PRSA is also extracted as a nine-residue-sliding-window feature (the number of feature: 9), and is available at same web service as PSS [38].

D. Sampling Strategy

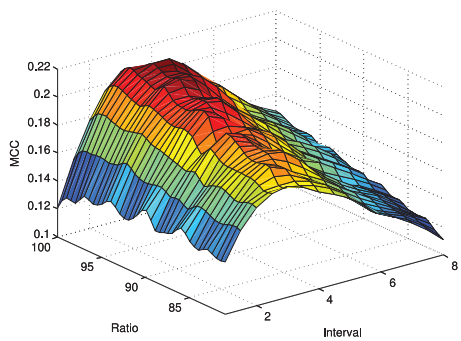
The extreme imbalance between the interacting and non-interacting residues, makes the choice of representative samples to be a big challenge. Hu *et al.* [40] proposed a supervision-based over-sampling algorithm, which utilizes the majority class information to guide the up-sample of minority class. However, the up-sampling method is computationally expensive, and has a risk of over fitting. For down-sampling, the most simple and effective heuristic method is random sampling. However, random sampling lacks the stability. In this paper, a new down-sampling strategy is proposed. The approach consists of the following 4 steps: Step 1, a tree is constructed using the raw training dataset as



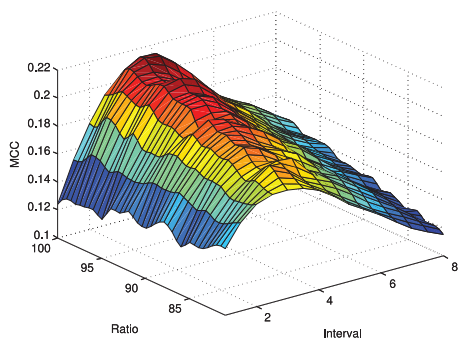
(a)



(b)



(c)



(d)

Fig. 5. The same "Ratio" and "Interval" are accepted in 2, 6, 10, and 14 "Cluster," the rangers of "Ratio" and "Interval" are [81, 100](%) and [1, 8] respectively with the step 1(%). The Dset186 is divided into 6 parts with 31 proteins respectively. The data above are the average of 6-fold cross validation. (a) MCC in 2 clusters, (b) MCC in 6 clusters, (c) MCC in 10 clusters, (d) MCC in 14 clusters.

the root; Step 2, the tree is grown using selected features; Step 3, the subsets in the leaves are clustered by K-means; Step 4, all the samples are sorted based on their distances to the corresponding cluster centers, and the training samples are then selected at a fixed interval (For example, given 100 points, 1st, 4th, 7th, 10th, ..., 97th, 100th point will be selected at a 3-interval where the points are sorted based on their closeness to the corresponding cluster centers). Note that only the negative samples are selected using the proposed sampling strategy (or samples in the majority class of a binary classification problem). All the positive samples (or samples in the minority class) will be included. The details of our proposed approach is shown in Algorithm 1.

Algorithm 1 PETs sampling algorithm

Input:

M: the raw dataset with feature vectors;

$m_{cluster}$: the number of cluster;

r: the ratio of sample considering from each center of cluster;

$n_{interval}$: the interval between each sampling;

$features_{divided}$: the list of features used for dividing raw dataset;

Output:

$Dataset_{training}$: the set of final training data

$Dataset_{sub} = \text{list}()$
for $feature$ in $features_{divided}$ **do**
 if $Dataset_{sub} = \text{None}$ **then**
  $Dataset_{sub}$ appends M·divided($feature$)
 else
  $Dataset_{temp} = Dataset_{sub}[:]$
  $Dataset_{sub} = \text{list}()$
  for $Dataset$ in $Dataset_{temp}$ **do**
   $Dataset_{sub}$ appends $Dataset$·divided($feature$)
  end for
 end if
end for
for $Dataset$ in $Dataset_{sub}$ **do**
 for $feature$ in $features_{divided}$ **do**
  del $Dataset[sample] \cdot features[feature]$
 end for
 $clf = \text{k-means}(m_{cluster})$
 $Dataset \cdot neg[sample] \cdot label, Dataset \cdot neg[sample] \cdot dist = clf \cdot \text{fit}(Dataset \cdot neg)$
 $clusters = \text{dict}()$
 for $sample$ in $Dataset \cdot neg$ **do**
  if not $clusters \cdot \text{has\_key}(sample \cdot label)$ **then**
   $clusters[sample \cdot label] = \text{list}()$
  end if
  $clusters[sample \cdot label]$ appends $sample$
 end for
 for $label$ in $clusters$·keys() **do**
  $clusters[label] \cdot \text{sort}(\text{key} = sample \cdot dist, \text{reverse} = \text{False})$
  $count = 0$
  for $index$ in xrange(r*len($clusters[label]$)) **do**
   if $count == n_{interval}$ **then**
    $count = 0$

32: $Dataset_{training}$ appends $M[clusters[label][index] \cdot name]$
33: **else**
34: $count+ = 1$
35: **end if**
36: **end for**
37: **end for**
38: **end for**
39: **return** $Dataset_{training}$

The aforementioned sampling strategy is a type of cluster sampling (or stratified sampling) where the clusters are obtained using K-means clustering. In particular, PSS and PSA, which have low-dimensional labels, are first used to segment the raw training dataset and K-means clustering, is then applied on the subsets. The idea of feature-based segmentation was motivated from the work in Folkman *et al.* [41]. However, in our work, we do not use the subsets to construct multiple models. The detail of feature-based segmentation is shown in Fig. 2. The negative samples are selected under each pattern (a pattern means a residue with a combination of PSS and PSA, e.g., PSS is α-helix while PSA is exposed or PSA is "intermediate"). From Fig. 2, we observe that the data distribution in layer 2 (in the middle) are quite similar; however, in layer 3 (on the right), the distributions of positive samples are different. This shows that the data segmentation is effective.

E. Extremely Randomized Trees

Extremely randomized trees (ETs) is a tree-based ensemble methods. ETs constructs an ensemble of unpruned decision or regression trees in a top-down manner. Different from Random Forests (RFs) [42], [43], the two primary innovations of ETs are: 1) the cut-points are selected randomly to divide nodes; 2) the decision trees are constructed by the whole training dataset rather than the replica generating via Bootstrap [44]. In this paper, we employs scikit-learn (a machine learning package in python) to implement ETs. Scikit-learn supports simple and efficient tools for data mining and data analysis, and is available at http://scikit-learn.org/stable/index.html [45].

F. Evaluation Criteria

The accuracy alone is not sufficient to evaluate a predictor. For the objectivity and effectiveness of evaluation, the concepts of confusion matrix and Receiver Operating Characteristics (ROC curve is drawn using the confusion matrix and the Area Under an ROC Curve (AUC)) are used to compare our method with other alternative algorithms. The confusion matrix and relevant evaluation index are illustrated in Fig. 3 [46].

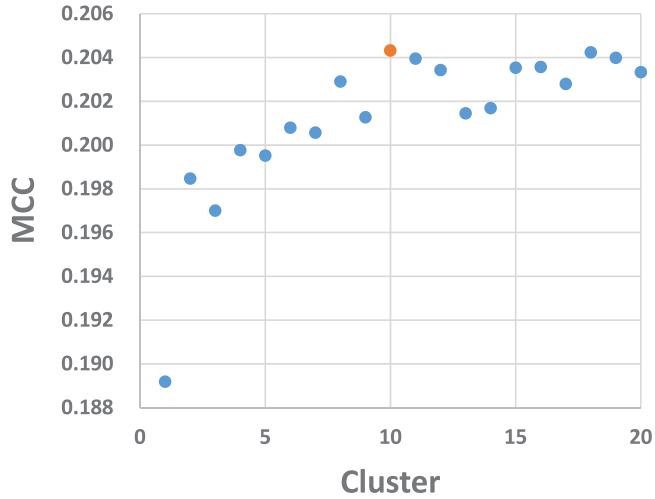


Fig. 6. The performance of our sampling strategy in different clusters.

True Positive Rate (TPR, Sensitivity, Recall), True Negative Rate (TNR, Specificity), Positive Predictive Value (PPV, Precision), Accuracy (ACC), Matthews Correlation Coefficient (MCC), and F-measure, are then defined as follows. Among these criteria, MCC, which is a correlation coefficient between the observed and predicted binary classifications, is the most important criterion in protein-protein interaction predictions. (See equations at the bottom of the page.)

The stability of a sampling strategy is also an evaluation objective in this paper. The most simple and effective method to measure the fluctuation is calculating the variance of each criteria above. The variance of a discrete random variable is calculated below:

$$\sigma_X^2 = E[X^2] - (E[X])^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2, \tag{8}$$

where μ means the average of random variable X of a set n.

$$TPR = \frac{TP}{TP + FN}; \tag{2}$$

$$TNR = \frac{TN}{TN + FP}; \tag{3}$$

$$PPV = \frac{TP}{TP + FP}; \tag{4}$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}; \tag{5}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}; \tag{6}$$

$$F\text{-measure} = \frac{2 \times (PPV \times TPR)}{PPV + TPR}; \tag{7}$$

TABLE III
THE PERFORMANCE OF SAMPLING METHODS IN DTESTSET72 AND PDBTESTSET164[1]

Dataset	Strategy	Recall(%)	Specificity(%)	Precision(%)	Accuracy(%)	**MCC(%)**	F-measure(%)
Dtestset72	PETs	65.1	63.8	19.0	63.9	**18.9**	29.4
		0.480	0.189	0.044	0.135	0.211	0.086
	Strategy II(1:1)	65.4	63.2	18.8	63.4	18.6	29.2
		0.897	0.632	0.050	0.247	0.258	0.102
	Strategy I(1:1)	65.5	63.1	18.8	63.4	18.7	29.2
		1.321	0.717	0.065	0.441	0.276	0.106
	Strategy I(1:1.1)	64.7	64.0	19.0	64.1	18.8	29.4
		1.127	0.497	0.066	0.311	0.323	0.130
PDBtestset164	PETs	60.4	61.0	25.2	60.9	**16.5**	35.5
		0.328	0.158	0.031	0.078	0.133	0.056
	Strategy II(1:1)	60.8	60.5	25.1	60.6	**16.5**	35.5
		0.537	0.403	0.035	0.178	0.122	0.047
	Strategy I(1:1)	60.1	61.2	25.2	61.0	**16.5**	35.5
		1.087	0.665	0.052	0.270	0.200	0.082
	Strategy I(1:1.1)	61.4	59.9	25.0	60.2	16.4	35.5
		0.950	0.631	0.052	0.279	0.212	0.080

[1] The second line in the result of each strategy is the variance of the corresponding criterion.

III. RESULTS AND DISCUSSION

A. Validation of Features

Four sequence-based protein features are used in our approach: Position Specific Scoring Matrix (PSSM), Protein Second Structure (PSS), Predicted Solvent Accessibility (PSA), and Predicted Relative Solvent Accessibility (PRSA). Our sampling strategy is feature-based, which means our sampling method could not work without PSS and PSA. So the random sampling, which maintains an equal distribution of positive and negative in each protein, will be used to validate the features in Section [17]. Leave-One-Out Cross Validation (LOOCV) is used to assess the performance of each feature under Dset186 with the definition of Murakami *et al.* [16]. ETs is used as the classifier. Table I presents the evaluation results of each combination.

Each protein in Dset186 is left out, and 186 trials are run in each LOOCV. To compare with LORIS, the results in Table I are obtained by averaging 3 trials. The average of cumulative hydropathy (ACH) used in LORIS does not have the good performance in our prediction model, so ACH is removed [17], [47]. In general, the performance measure becomes better with a good balance between Recall and Specificity when a new feature is added or a wider sliding window is used. We found an interesting phenomenon here: F-measures in the first line are not equal to the equation $(2 \times (\text{Recall} \times \text{Specificity}))/(\text{Recall} + \text{Specificity})$, besides LORIS. That is because, the results in the first lines are calculated using the average, those in the second lines are calculated using the cumulative confusion matrix. They have similar trends, but with different values. The special LOOCV is used here for verifying the features, and every round of LOOCV leaves out different proteins with different lengths. Each round will lose some information, however, the cumulative confusion matrix could retain much information in each round of training and testing. To compare with LORIS, the results of two calculating methods are displayed. Here, we also replace features used in LORIS with our features and we achieve better results than LORIS.

Table II shows the performance of LORIS's and our features in Dset186 and PDBtestset164 using ETs and L_1-logreg, which are the classifiers of PETs and LORIS.

Fig. 4 shows the weight of each feature in PSSM. "Top" means the best in PSSM, not in all features. In fact, PRSA and PSA are more effective in our model.

B. Performance of Sampling Strategy

In this paper, a new sampling method is proposed to improve the stability and accuracy. The proposed sampling needs 3 input parameters: Ratio, Interval, and Cluster. "Ratio" is the proportion of negative samples, "Interval" is a fixed interval of selecting a sample from the sorted list, and "Cluster" represents the number of clusters when using K-means. MCC is used as the main evaluation criterion along with the balance between Recall and Specificity. Fig. 5 shows the relationship between "Ratio" and "Interval."

Obviously, MCC reaches the peak when r is more than 90% and n_{interval} is in [3, 4]. However, it is the optimal MCC without considering the balance between Recall and Specificity. A suitable balance ($\text{Recall} = 62.9\%$ and $\text{Specificity} = 63.0\%$) can be found when $r = 100\%$ and $n_{\text{interval}} = 4$. Once "Ratio" and the "Interval" are determined, then the optimal number of cluster (m_{cluster}) can be selected using them. The trend of m_{cluster} is displayed in Fig. 6.

The scatter-plot displays the actual variation of MCC across different clusters. From Fig. 6, we can find that, MCC reaches the peak at $m_{\text{cluster}} = 10$, which has a suitable balance between Recall and Specificity.

We compare our sampling method with two other alternative strategies: 1) Strategy I: For this strategy, all amino acids in each protein are gathered in a set, and negative samples are then selected based on the number of positive residues where the ratios of 1:1 and 1:1.1 (positive : negative) will be used. 2) Strategy II: This strategy is used by Dhole *et al.* [17], which maintains the ratio of 1:1 between positive and negative residues and keeps this balance in each protein. In particular, if the positive amino acids are more than the negative ones in a protein, the positive samples will be selected based on the number of the negative residues. Table III shows the results about the comparison. The first line represents the value of each criterion and the ones in

TABLE IV
THE PERFORMANCE OF PETS AND LORIS ON DIFFERENT DTESTSET72[1]

	Recall(%)	Specificity(%)	Precision(%)	Accuracy(%)	**MCC(%)**	F-measure(%)
Dset-Murakami						
PETs	65.1	63.8	19.0	63.9	**18.9**	29.4
	0.480	0.189	0.044	0.135	0.211	0.086
LORIS	61.2	62.5	17.6	62.4	15.5	27.3
	0.290	0.147	0.024	0.103	0.119	0.043
PETs-L_1-logreg	52.1	70.2	18.6	68.2	15.3	27.4
	0.302	0.128	0.040	0.091	0.144	0.069
LORIS	63.1	61.0	23.8	61.4	17.7	32.4
PSIVER	46.5	69.3	25.0	66.1	13.5	32.5
ISIS	35.0	76.2	21.0	70.9	9.1	26.3
SPPIDER	45.4	64.7	20.4	61.7	8.1	24.6
Dset-ASAChange						
PETs	64.3	65.0	30.7	64.9	**23.6**	41.5
	0.486	0.186	0.072	0.099	0.266	0.144
LORIS	59.7	64.5	28.8	63.6	19.6	38.9
	0.248	0.212	0.054	0.111	0.157	0.063
PSIVER	63.7	53.7	24.8	55.6	13.7	35.7
Dset-ANDistance						
PETs	65.5	64.4	26.5	64.6	**22.6**	37.8
	0.474	0.198	0.088	0.139	0.341	0.145
LORIS	60.4	64.4	24.9	63.7	18.7	35.3
	0.404	0.194	0.059	0.119	0.225	0.092
PSIVER	64.1	53.2	21.1	55.0	12.8	31.9
Dset-AVWRDistance						
PETs	65.8	65.4	23.2	65.5	**22.0**	34.3
	0.452	0.214	0.060	0.140	0.225	0.100
LORIS	60.4	65.0	21.5	64.4	18.0	31.7
	0.462	0.298	0.070	0.203	0.261	0.111
PSIVER	63.7	52.6	17.6	54.1	11.2	27.6
Dset-PIADA						
PETs	65.0	64.7	27.2	64.8	**22.7**	38.4
	0.461	0.183	0.062	0.108	0.252	0.105
LORIS	59.8	64.1	25.3	63.3	18.3	35.5
	0.347	0.267	0.065	0.154	0.199	0.082
PSIVER	63.5	53.2	21.6	54.9	12.5	32.2

[1] The second line in the result of each predictor is the variance of criteria above.

TABLE V
THE PERFORMANCE OF PETS AND LORIS ON DIFFERENT PDBTESTSET164[1]

	Recall(%)	Specificity(%)	Precision(%)	Accuracy(%)	**MCC(%)**	F-measure(%)
Dset-ASAChange						
PETs	60.4	61.0	25.2	60.9	**16.5**	35.5
	0.328	0.158	0.031	0.078	0.133	0.056
LORIS	50.3	62.1	22.4	60.0	9.6	30.9
	0.253	0.166	0.019	0.079	0.076	0.032
PSIVER	45.8	62.4	21.2	59.4	6.4	28.9
Dset-ANDistance						
PETs	61.7	60.2	22.1	60.4	**16.0**	32.6
	0.402	0.118	0.028	0.069	0.159	0.060
LORIS	49.3	62.4	19.4	60.3	8.7	27.8
	0.274	0.137	0.021	0.076	0.088	0.039
PSIVER	45.4	62.1	18.2	59.5	5.6	26.0
Dset-AVWRDistance						
PETs	62.1	60.1	19.1	60.3	**15.1**	29.2
	0.360	0.153	0.019	0.088	0.110	0.041
LORIS	48.4	63.2	16.7	61.2	8.1	24.8
	0.242	0.243	0.022	0.152	0.087	0.034
PSIVER	45.3	61.9	15.4	59.7	5.0	23.0
Dset-PIADA						
PETs	61.5	60.3	22.5	60.5	**16.1**	32.9
	0.371	0.155	0.023	0.080	0.130	0.050
LORIS	49.7	61.8	19.6	59.9	8.6	28.1
	0.360	0.202	0.017	0.105	0.124	0.061
PSIVER	45.4	62.1	18.5	59.4	5.6	26.3

[1] The second line in the result of each predictor is the variance of corresponding criterion.

brackets represent the variance. Each strategy is run 100 times for evaluating the stability.

All three random sampling methods have a good balance between Recall and Specificity. Although our proposed sam-

pling strategy has similar accuracy values with other baseline methods, it is more stable.

C. Comparison of Stability in Datasets of Different Definitions

The datasets are redefined using the program Mihel *et al.* provided [30]. More details can be found in Appendix A.

For comparison, we implemented LORIS [17]. In particular, PSSM and ACH are normalized using the sigmoid function and PRSA is extracted from SANN. The scores next to labels (E, B, I) in SANN's reports are used as the features in our reproduction.

Tables IV and V present the results on the datasets with different definitions. The performance measure is displayed on the first line and the second line is the variance. The results are calculated based on the cumulative confusion matrix with 100 runs.

Dhole *et al.* also evaluated their predictor on PDBtestset164; however, they used the model which was trained by original Dset186 to test on PDBtestset164 (labeled by PIADA). Although Dset-ASAChange and Dset-Murakami used the same method to label the dataset, in fact, 0.24% of their labels are different.

Notably, LORIS has similar variance with PETs. Generally, the more simple the model is, the more stable (i.e., with less variance) it would be [44], [48], [49]. LORIS uses L_1-logreg [18] as the classifier while PETs uses ETs. Table IV also shows the variance of PETs using L_1-logreg [18]. There is little difference between the variances of PETs and LORIS.

D. Performance on Different Types of Proteins

Dset186, Dtestset72, and PDBtestset164 contain a variety of proteins, which are different in amino acid compositions, sequence length, physicochemical properties, and so on. Fig. 7 shows the performance of PETs with different sequence lengths. In Fig. 7, each point indicates a protein in Dtestset72 or PDBtestset164. The X-axis shows the length of protein sequence, and the Y-axis presents the MCC, which is evaluated using PETs to predict the interacting amino acids in this protein.

From Fig. 7, we can find that, PETs has good performance in proteins with different sequence lengths, especially in those short proteins. Since long proteins are rare in nature, a few long proteins are available in our experiment. Table VI shows the performance of PETs with different amino acids.

IV. Conclusion

Identifying the regions of interaction in proteins is difficult, and classifying each residue is even more challenging. Especially in practical applications, the stability and accuracy of classification are equally important. In this paper, we proposed PETs (**P**redictor of Protein-Protein interaction sites based on **E**xtremely-randomized **T**ree**s**) to improve the accuracy while maintaining the stability. A new sampling strategy is proposed to solve this particular problem: a) the raw training dataset is divided into several subsets, and b) the samples of the training dataset are selected from each cluster of subsets using K-means. The source code and toolkit are available at https://github.com/BinXia/PETs.

There are some research questions. First, the residues, which are the first and last 4 residues of a protein, have a higher proportion of interaction than those in the whole sequence of proteins. Can a special model be constructed to fit these edge residues? Second, can the effective feature be extracted to better identify the interacting residues? Finally, can some sampling strategies be applied to the residue-scale PPI classification?

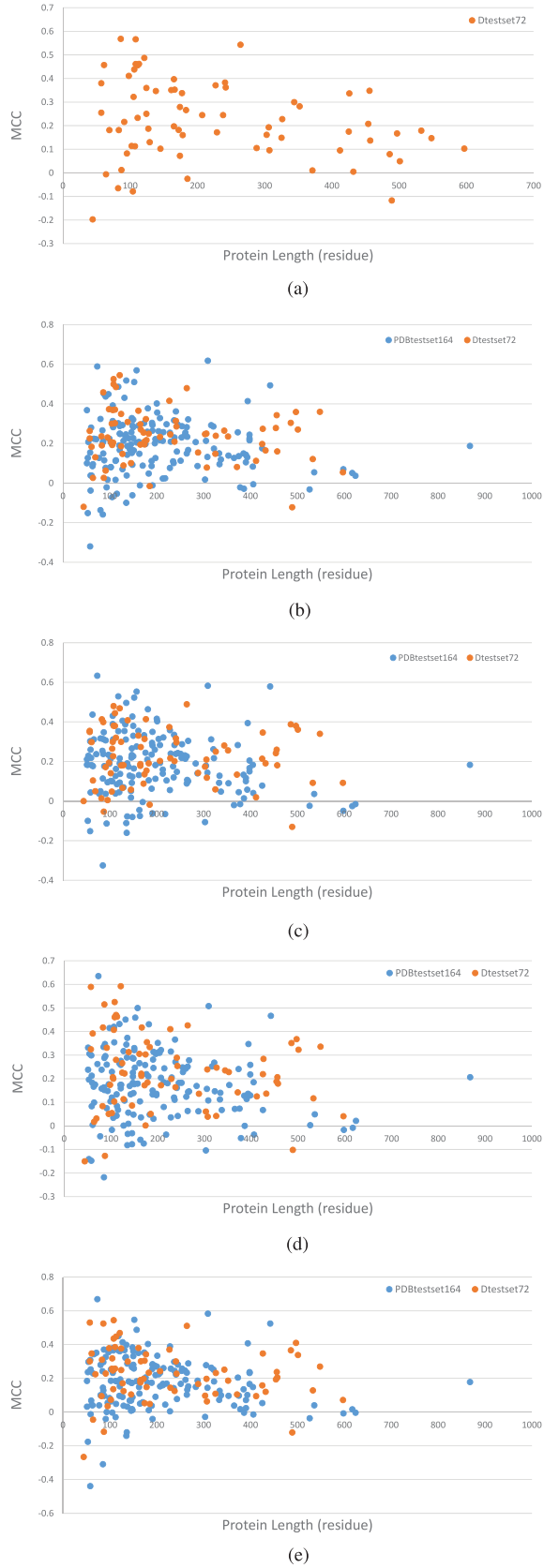


Fig. 7. The performance of each protein in Dtestset72 and PDBtestset164 under different definitions. (a) Dset-Murakami, (b) Dset-ASAChange, (c) Dset-ANDistance, (d) Dset-AVWRDistance, (e) Dset-PIADA.

TABLE VI
THE PERFORMANCE OF PETS IN DIFFERENT AMINO ACID

Residue	MCC(%)		F-measure(%)	
	Dtestset72	PDBtestset164	Dtestset72	PDBtestset164
A	20.5	14.7	34.5	32.1
C	23.7	16.9	37.2	36.3
D	24.4	16.0	38.0	33.3
E	20.2	13.6	34.8	30.8
F	25.6	17.5	38.0	35.1
G	23.4	16.4	37.6	32.2
H	22.9	10.8	38.1	26.0
I	21.3	17.4	34.6	32.4
K	20.9	16.8	34.1	34.0
L	20.4	13.5	35.2	30.7
M	23.7	19.5	36.7	34.0
N	25.5	14.7	40.3	31.5
P	21.4	16.6	37.3	32.6
Q	21.6	16.7	36.5	32.5
R	18.8	18.0	33.3	34.7
S	22.8	17.5	39.5	34.5
T	22.0	17.8	36.0	34.0
V	21.4	16.3	34.6	32.5
W	18.4	14.1	32.0	29.1
Y	26.2	15.9	38.7	32.3

TABLE VII
THE INTERFACE DISTRIBUTION OF DSET-MURAKAMI

Dset-Murakami				
Residues	interacting	non-interacting	total	ratio
Dset186-total	5517	30702	36219	15.23%
Dset186-valid	5241	29490	34731	15.09%
Dtestset72-total	1867	14273	16140	11.57%
Dtestset72-valid	1796	13776	15572	11.53%

TABLE VIII
THE INTERFACE DISTRIBUTION OF DSET-ASACHANGE

Dset-ASAChange				
Residues	interacting	non-interacting	total	ratio
Dset186-total	5551	30668	36219	15.33%
Dset186-valid	5274	29457	34731	15.19%
Dtestset72-total	3128	13012	16140	19.38%
Dtestset72-valid	3020	12552	15572	19.39%
PDBtestset164-total	6096	27585	33681	18.10%
PDBtestset164-valid	5780	26589	32369	17.86%

APPENDIX A
THE DETAIL OF DATASETS

In this section, we present the details of each dataset. Tables VII, VIII, IX, X, and XI show the detailed interface distribution of Dset-ASAChange, Dset-Murakami, Dset-ANDistance, Dset-AVWRDistance, and Dset-PIADA respectively. Notably, "total" means the proteins with first and last 4 residues while "valid" is without those residues.

Tables XII, XIII, and XIV present the repetition ratio between different definitions of datasets. MK, ASAC, AND, AVWRD, and PIADA are abbreviations of Murakami, ASAChange, ANDistance, AVWRDistance, and PIADA, respectively.

TABLE IX
THE INTERFACE DISTRIBUTION OF DSET-ANDISTANCE

Dset-ANDistance				
Residues	interacting	non-interacting	total	ratio
Dset186-total	4805	31414	36219	13.27%
Dset186-valid	4573	30158	34731	13.17%
Dtestset72-total	2650	13490	16140	16.42%
Dtestset72-valid	2554	13018	15572	16.40%
PDBtestset164-total	5286	28395	33681	15.69%
PDBtestset164-valid	5024	27345	32369	15.52%

TABLE X
THE INTERFACE DISTRIBUTION OF DSET-AVWRDISTANCE

Dset-AVWRDistance				
Residues	interacting	non-interacting	total	ratio
Dset186-total	4107	32112	36219	11.34%
Dset186-valid	3898	30833	34731	11.22%
Dtestset72-total	2212	13928	16140	13.71%
Dtestset72-valid	2129	13443	15572	13.67%
PDBtestset164-total	4483	29198	33681	13.31%
PDBtestset164-valid	4275	28094	32369	13.21%

TABLE XI
THE INTERFACE DISTRIBUTION OF DSET-PIADA

Dset-PIADA				
Residues	interacting	non-interacting	total	ratio
Dset186-total	4909	31310	36219	13.55%
Dset186-valid	4671	30060	34731	13.45%
Dtestset72-total	2726	13414	16140	16.89%
Dtestset72-valid	2628	12944	15572	16.88%
PDBtestset164-total	5346	28256	33602	15.91%
PDBtestset164-valid	5080	27210	32290	15.73%

TABLE XII
THE REPETITION RATIO OF DSET186 BETWEEN DIFFERENT DEFINITIONS

	Dset186				
	MK	ASAC	AND	AVWRD	PIADA
MK	100.00%	99.76%	97.61%	95.97%	97.75%
ASAC	99.76%	100.00%	97.48%	95.86%	97.61%
AND	97.61%	97.48%	100.00%	98.07%	99.36%
AVWRD	95.97%	95.86%	98.07%	100.00%	97.79%
PIADA	97.75%	97.61%	99.36%	97.79%	100.00%

TABLE XIII
THE REPETITION RATIO OF DTESTSET72 BETWEEN DIFFERENT DEFINITIONS

	Dtestset72				
	MK	ASAC	AND	AVWRD	PIADA
MK	100.00%	92.11%	91.56%	91.08%	91.44%
ASAC	92.11%	100.00%	96.54%	94.24%	96.89%
AND	91.56%	96.54%	100.00%	97.29%	99.05%
AVWRD	91.08%	94.24%	97.29%	100.00%	96.82%
PIADA	91.44%	96.89%	99.05%	96.82%	100.00%

Notably, the residues of PDBtestset164 in Dset-PIADA is different with others. PIADA removed 79 residues (0.23%) with uncertain label in some proteins, since not enough information can be provided by PIADA. For comparison with different

TABLE XIV
THE REPETITION RATIO OF PDBTESTSET164 BETWEEN DIFFERENT DEFINITIONS

	PDBtestset164				
	MK	ASAC	AND	AVWRD	PIADA
MK	-	-	-	-	-
ASAC	-	100.00%	97.05%	95.06%	97.23%
AND	-	97.05%	100.00%	97.62%	99.33%
AVWRD	-	95.06%	97.62%	100.00%	97.41%
PIADA	-	97.23%	99.33%	97.41%	100.00%

definitions, the residues, which PIADA removed, would be removed from other definitions.

REFERENCES

[1] T. Pawson and P. Nash, "Assembly of cell regulatory systems through protein interaction domains," *Science*, vol. 300, no. 5618, pp. 445–452, 2003.

[2] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki, "A comprehensive two-hybrid analysis to explore the yeast protein interactome," *Proc. Natl. Acad. Sci.*, vol. 98, no. 8, pp. 4569–4574, 2001.

[3] L. Giot *et al.*, "A protein interaction map of drosophila melanogaster," *Science*, vol. 302, no. 5651, pp. 1727–1736, 2003.

[4] A. Baspinar, E. Cukuroglu, R. Nussinov, O. Keskin, and A. Gursoy, "Prism: A web server and repository for prediction of protein-protein interactions and modeling their 3d complexes," *Nucl. Acids Res.*, vol. 42, no. W1, pp. W285–W289, 2014.

[5] A. Birlutiu and T. Heskes, "Using topology information for protein-protein interaction prediction," in *Pattern Recognition in Bioinformatics*. New York: Springer, 2014, pp. 10–22.

[6] Z. Dong, K. Wang, T. K. L. Dang, M. Gültas, M. Welter, T. Wierschin, M. Stanke, and S. Waack, "Crf-based models of protein surfaces improve protein-protein interaction site predictions," *BMC Bioinformat.*, vol. 15, no. 1, p. 277, 2014.

[7] M. Ohue, Y. Matsuzaki, N. Uchikoga, T. Ishida, and Y. Akiyama, "Megadock: An all-to-all protein-protein interaction prediction system using tertiary structure data," *Protein Peptide Lett.*, vol. 21, no. 8, p. 766, 2014.

[8] Y. Zhang, E. Zeng, T. Li, and G. Narasimhan, "Weighted consensus clustering for identifying functional modules in protein-protein interaction networks," in *Proc. IEEE Int. Conf. Mach. Learn. Appl. (ICMLA'09)*, 2009, pp. 539–544.

[9] D. Wang, M. Ogihara, E. Zeng, and T. Li, "Combining gene expression profiles and protein-protein interactions for identifying functional modules," in *Proc. IEEE 11th Int. Conf. Mach. Learn. Appl. (ICMLA)*, 2012, vol. 1, pp. 114–119.

[10] J. Shen, J. Zhang, X. Luo, W. Zhu, K. Yu, K. Chen, Y. Li, and H. Jiang, "Predicting protein-protein interactions based only on sequences information," *Proc. Natl. Acad. Sci.*, vol. 104, no. 11, pp. 4337–4341, 2007.

[11] Z.-H. You, Y.-K. Lei, L. Zhu, J. Xia, and B. Wang, "Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis," *BMC Bioinformat.*, vol. 14, no. Suppl. 8, p. S10, 2013.

[12] M. Šikić, S. Tomić, and K. Vlahoviček, "Prediction of protein-protein interaction sites in sequences and 3d structures by random forests," *PLoS Comput. Biol.*, vol. 5, no. 1, p. e1000278, 2009.

[13] P. Fariselli, F. Pazos, A. Valencia, and R. Casadio, "Prediction of protein-protein interaction sites in heterocomplexes with neural networks," *Eur. J. Biochem.*, vol. 269, no. 5, pp. 1356–1361, 2002.

[14] A. Porollo and J. Meller, "Prediction-based fingerprints of protein-protein interactions," *Proteins: Struct., Funct., Bioinformat.*, vol. 66, no. 3, pp. 630–645, 2007.

[15] Y. Ofran and B. Rost, "Isis: Interaction sites identified from sequence," *Bioinformatics*, vol. 23, no. 2, pp. e13–e16, 2007.

[16] Y. Murakami and K. Mizuguchi, "Applying the naïve Bayes classifier with kernel density estimation to the prediction of protein-protein interaction sites," *Bioinformatics*, vol. 26, no. 15, pp. 1841–1848, 2010.

[17] K. Dhole, G. Singh, P. P. Pai, and S. Mondal, "Sequence-based prediction of protein-protein interaction sites with l1-logreg classifier," *J. Theoretical Biol.*, vol. 348, pp. 47–54, 2014.

[18] K. Koh, S.-J. Kim, and S. P. Boyd, "An interior-point method for large-scale l1-regularized logistic regression," *J. Mach. Learn. Res.*, vol. 8, no. 8, pp. 1519–1555, 2007.

[19] S. Jones and J. M. Thornton, "Prediction of protein-protein interaction sites using patch analysis," *J. Mol. Biol.*, vol. 272, no. 1, pp. 133–143, 1997.

[20] S. Jones and J. M. Thornton, "Analysis of protein-protein interaction sites using surface patches," *J. Mol. Biol.*, vol. 272, no. 1, pp. 121–132, 1997.

[21] S. J. Hubbard and J. M. Thornton, Naccess. ver. 2.1.1, 1993, Computer Program, Dept. Biochem. Mol. Biol., University College London, U.K.

[22] B. Lee and F. M. Richards, "The interpretation of protein structures: Estimation of static accessibility," *J. Mol. Biol.*, vol. 55, no. 3, pp. 379–IN4, 1971.

[23] X. Gallet, B. Charloteaux, A. Thomas, and R. Brasseur, "A fast method to predict protein interaction sites from sequences," *J. Mol. Biol.*, vol. 302, no. 4, pp. 917–926, 2000.

[24] J. R. Bock and D. A. Gough, "Predicting protein-protein interactions from primary structure," *Bioinformatics*, vol. 17, no. 5, pp. 455–460, 2001.

[25] C. Zeng, T. Li, L. Shwartz, and G. Y. Grabarnik, "Hierarchical multi-label classification over ticket data using contextual loss," in *Proc. IEEE Netw. Op. Manage. Symp. (NOMS)*, 2014, pp. 1–8.

[26] C. Zeng *et al.*, "Fiu-miner: A fast, integrated, and user-friendly system for data mining in distributed environment," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 1506–1509.

[27] L. Zheng *et al.*, "Applying data mining techniques to address critical process optimization needs in advanced manufacturing," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 1739–1748.

[28] F. C. Bernstein, T. F. Koetzle, G. J. Williams, E. F. Meyer, M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi, "The protein data bank," *Eur. J. Biochem.*, vol. 80, no. 2, pp. 319–324, 1977.

[29] G. Singh, K. Dhole, P. P. Pai, and S. Mondal, "Springs: Prediction of protein-protein interaction sites using artificial neural networks," PeerJ PrePrints, Tech. Rep., 2014.

[30] J. Mihel, M. Šikić, S. Tomić, B. Jeren, and K. Vlahoviček, "Psaia-protein structure and interaction analyzer," *BMC Struct. Biol.*, vol. 8, no. 1, p. 21, 2008.

[31] Y. Ofran and B. Rost, "Predicted protein-protein interaction sites from local sequence information," *FEBS Lett.*, vol. 544, no. 1, pp. 236–239, 2003.

[32] A. S. Aytuna, A. Gursoy, and O. Keskin, "Prediction of protein-protein interactions by combining structure and sequence conservation in protein interfaces," *Bioinformatics*, vol. 21, no. 12, pp. 2850–2855, 2005.

[33] D.-J. Yu, J. Hu, Y. Huang, H.-B. Shen, Y. Qi, Z.-M. Tang, and J.-Y. Yang, "Targetatpsite: A template-free method for atp-binding sites prediction with residue evolution image sparse representation and classifier ensemble," *J. Comput. Chem.*, vol. 34, no. 11, pp. 974–985, 2013.

[34] C. Zeng, H. Li, H. Wang, Y. Guang, C. Liu, T. Li, M. Zhang, S.-C. Chen, and N. Rishe, "Optimizing online spatial data analysis with sequential query patterns," in *Proc. IEEE 15th Int. Conf. Inf. Reuse Integr. (IRI)*, 2014, pp. 253–260.

[35] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped blast and psi-blast: A new generation of protein database search programs," *Nucl. Acids Res.*, vol. 25, no. 17, pp. 3389–3402, 1997.

[36] C. Camacho, G. Coulouris, V. Avagyan, N. Ma, J. Papadopoulos, K. Bealer, and T. L. Madden, "Blast+: Architecture and applications," *BMC Bioinformat.*, vol. 10, no. 1, p. 421, 2009.

[37] D. Yu, J. Hu, J. Yang, H. Shen, and J. Tang, "Designing template-free predictor for targeting protein-ligand binding sites with classifier ensemble and spatial clustering," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 10, no. 4, pp. 994–1008, 2013.

[38] J. Cheng, A. Z. Randall, M. J. Sweredoski, and P. Baldi, "Scratch: A protein structure and structural feature prediction server," *Nucl. Acids Res.*, vol. 33, no. suppl. 2, pp. W72–W76, 2005.

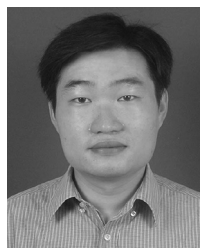
[39] K. Joo, S. J. Lee, and J. Lee, "Sann: Solvent accessibility prediction of proteins by nearest neighbor method," *Proteins: Struct., Funct., Bioinformat.*, vol. 80, no. 7, pp. 1791–1797, 2012.

[40] J. Hu, X. He, D.-J. Yu, X.-B. Yang, J.-Y. Yang, and H.-B. Shen, "A new supervised over-sampling algorithm with application to protein-nucleotide binding residue prediction," *PloS One*, vol. 9, no. 9, p. e107676, 2014.

[41] L. Folkman, B. Stantic, and A. Sattar, "Feature-based multiple models improve classification of mutation-induced stability changes," *BMC Genomics*, vol. 15, no. 4, pp. 1–11, 2014.
[42] Q.-F. Zhou, H. Zhou, Y.-P. Ning, F. Yang, and T. Li, "Two approaches for novelty detection using random forest," *Expert Syst. With Appl.*, vol. 42, no. 10, pp. 4840–4850, 2015.
[43] Q. Zhou, Y. Ning, Q. Zhou, L. Luo, and J. Lei, "Structural damage detection method based on random forests and data fusion," *Struct. Health Monitor.*, vol. 12, no. 1, pp. 48–58, 2013.
[44] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006.
[45] L. Breiman *et al.*, "Statistical modeling: The two cultures (with comments and a rejoinder by the author)," *Stat. Sci.*, vol. 16, no. 3, pp. 199–231, 2001.
[46] T. Fawcett, "An introduction to roc analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.
[47] J. Kyte and R. F. Doolittle, "A simple method for displaying the hydropathic character of a protein," *J. Mol. Biol.*, vol. 157, no. 1, pp. 105–132, 1982.
[48] D. Freedman, *Statistical Models: Theory and Practice*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[49] O. Bousquet, S. Boucheron, and G. Lugosi, "Introduction to statistical learning theory," in *Advanced Lectures on Machine Learning*. New York: Springer, 2004, pp. 169–207.

Bin Xia received his B.S. degree in electronic science and technology from Nanjing University of Science and Technology Zijin College, China, in 2012. Currently, he is working towards the Ph.D. 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.

Hong Zhang received the B.S. and M.S. degree in computer science from Nanjing University of Science and Technology (NUST), China. He is currently a full Professor at the School of Computer Science and Engineering, Nanjing University of Science and Technology. His current interests include confidence software technology and information security.



Qianmu Li received the B.S. degree and the Ph.D. degree in computer science from Nanjing University of Science and Technology, China, in 2001 and 2005, respectively. In 2012, he was an Academic Visitor at the Florida International University, USA. He is currently a full Professor at the School of Computer Science and Engineering, Nanjing University of Science and Technology. His current interests include data mining and information security. He is a member of the ACM and the CCF.

Tao Li received the Ph.D. degree in computer science from the Department of Computer Science, University of Rochester, Rochester, NY, USA, in 2004. He is currently a Professor with the School of Computing and Information Sciences, Florida International University, Miami, FL, USA. He is also a professor with the School of Computer Science & Technology, Nanjing University of Posts and Telecommunications (NJUPT), China. His research interests include data mining, computing system management, information retrieval, and machine learning. He received the U.S. National Science Foundation (NSF) CAREER Award and multiple IBM Faculty Research Awards.