Detection of outlier residues for improving interface prediction in protein hetero-complexes

Peng Chen, Limsoon Wong, Jinyan Li

Abstract—Sequence-based understanding and identification of protein binding interfaces is a challenging research topic due to the complexity in protein systems and the imbalanced distribution between interface and non-interface residues. This paper presents an outlier detection idea to address the redundancy problem in protein interaction data. The cleaned training data is then used for improving the prediction performance. We use three novel measures to describe the extent a residue is considered as an outlier in comparison to the other residues: the distance of a residue instance from the center instance of all residue instances of the same class label (Dist), the probability of the class label of the residue instance (PCL), and the importance of within-class and between-class (IWB) residue instances. Outlier scores are computed by integrating the three factors; instances with a sufficiently large score are treated as outliers and removed. The data sets without outliers are taken as input for a support vector machine (SVM) ensemble. The proposed SVM ensemble trained on input data without outliers performs better than that with outliers. Our method is also more accurate than many literature methods on benchmark data sets. From our empirical studies, we found that some outlier interface residues are truly near to non-interface regions, and some outlier non-interface residues are close to interface regions.

Index Terms—Outlier detection; protein-protein interaction; SVM ensemble.

1 Introduction

Outlier detection is to find patterns in data that do not conform to expected behaviors. These non-conforming patterns are often referred to as anomalies or outliers. Conceptually, an outlier is an observation that deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism [22], or alternately, an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data [23], [24]. Based on the availability of class label information, outlier detection techniques have three categories [8]: supervised outlier detection, which assumes the availability of an entire training data set with instances labeled as normal or anomaly class [43]; semi-supervised outlier detection, which operates in a semi-supervised mode, assuming that the training data has labeled instances for only the normal class [35]; unsupervised outlier detection, which operates in unsupervised mode, not requiring training data, and thus most widely applicable [18], [25]. Outliers may arise due to mechanical faults, changes in system behavior, human error, or other errors. Outliers in protein interface residues are not well investigated before. This work follows a general idea of outliers for the study of interface residue identification: Given a set of observations with class labels, find those that arouse suspicions, taking into account the class labels [4], [8], [22], [23], [24].

Kleywegt and Jones assigned those residues that are not located at the core regions on the Ramachandran plot as outliers [30]. They found that ∼91% of 3000 protein structures (from the Protein Data Bank before 1996) have up to 10% outliers [30]. Haliloglu *et al.* [21] presented an idea to define outlier residues by computing the shortest distance between HFV residues (high-frequency vibrating residue identified by the Gaussian network model [15]) and some conserved residues. A conserved residue is an outlier if the conserved residue does not overlap with any HFV residues at <7Å. Under this definition, about 3% of conserved residues are outliers [21]. They also presented some interesting explanations on how these outlying conserved residues occur: (a) the HFV residues may not belong to any binding region or to any folding core; (b) inaccuracies may exist in the multiple structural superpositions of conserved residues due to the presence of crystal interfaces in the data set; and (c) a residue may be conserved by a different reason, for example, by a specific functional

- *P. Chen is with the Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, P.R. China; and the Bioinformatics Research Center, School of Computer Engineering, Nanyang Technological University, Singapore 639798. E-mail: pchen1978@gmail.com;*
- *Limsoon Wong is with the School of Computing, National University of Singapore, Singapore 117417. E-mail: wongls@comp.nus.edu.sg.*
- *Jinyan Li is with the School of Computing, National University of Singapore, Singapore 117417; and the Advanced Analytics Institute, University of Technology Sydney, Australia. Corresponding author. E-mail: jinyan.li@uts.edu.au.*

interaction [21]. Similarly, protein-binding interface outliers in complexes can occur for two reasons: (a) some interface residues are physically near to non-interface residues in space which may bring up a confusion effect in the classification; (b) inaccuracies may also exist in the generation of the raw data of interacting residues. These two factors add complexity to the challenging protein-binding interface prediction problem.

Our work here aims to detect outliers respectively from interface residues and non-interface residues, and to remove them from the training step to purify the data. We expect that this outlier detection can improve the performance of interface prediction remarkably.

The interface prediction problem has received increasing attention since the pioneering work by Kini and Evans [29], which finds that proline is the most common residue in the flanking segments of interaction sites. There are roughly two main types of protein interface prediction: sequence-based or structure-based approach. Features used in sequence-based methods include residue composition and propensity [16], [27], hydrophobic scale [19], predicted structural features such as secondary structures [39], features extracted from multiple sequence alignment [20], [49], and so on [17]. On the other hand, some methods exploited structure-based properties including size of interfaces [27], [42], shape of interfaces [2], [26], [32], clustering of interface atoms [1], [14], B-factor [13], electrostatic potential [6], [13], spatial distribution of interface residues [1], [14], and others [47].

None of these methods considers outlier detection and removal to preprocess the training data. This work takes a new sequence-based approach to protein-binding interface prediction by using novel ideas to characterize outlier residues. We propose three measures to describe the extent to which a residue instance is likely to be an outlier compared to the others. The first measure is used to describe the distance of a residue instance from the center vector of all residue instances with the same class label (Dist). The second measure is the probability of the class label (PCL) of the residue instance. The third measure describes the importance of within-class and between-class (IWB) instances. An outlier score is then computed by integrating the three measures. Instances with a sufficiently large score are treated as outliers and are subsequently removed from the training data.

The resulting data sets without outliers are taken as input to a support vector machine ensemble to identify residues that are potentially interacting. As the data is highly imbalanced between interfacial residues and non-interfacial residues, we propose a stratification idea to split the large class of non-interfacial residues into equally-sized smaller parts to be trained with the class of interfacial residues. Results showed that our prediction method with outlier detection can perform better than that without the outlier detection. It achieves an MCC improvement of around 4.4% and F1 improvement by around 3.6%. We also found that some outlier interface residues are truly near to non-interface regions and, similarly, some outlier non-interface residues are close to interface regions.

2 Materials and Methods

2.1 Data Set

The complex data set used in this work was taken from our previous work [12], which contains 2499 protein chains in 737 complexes. Here, only those proteins in hetero-complexes with sequence identity $\leq 30\%$ were selected, and proteins and molecules with fewer than 30 residues were excluded from our data set. In addition, protein chains which are not available in HSSP database [44] were also removed. Moreover, accessible surface area (ASA) change is used to define interface residues. We used the PSAIA software [36] to compute the ASAs. A residue is considered to be an interface residue if the difference of its ASA between the unbound and bound forms is $>$1Å. Under this definition, there are 142410 interface residues (positive samples) and 374346 non-interface residues (negative samples). Thus, the positive samples account for only 27.56% in the total samples.

2.2 Feature vector representing a residue

For a given residue i in a protein chain, a sliding window with 19 residues is used to involve the association among its neighboring residues. The residue i is centered on the window. A novel encoding schema integrating hydrophobic scale and sequence profile is used to describe a residue. The sequence profile for one residue extracted from HSSP database [44] is then multiplied by the Kyte-Doolittle hydropathy scale [31]. For instance, the profile SP_k for residue k of the 19 residues and the Kyte hydropathy scale, KD, are both vectors with 1×20 dimensions. Thereafter, $MSK_k = SP_k \times KD$ for residue k represents the multiplication of the corresponding sequence profile by the Kyte hydropathy scale, whose jth element $MSK_k^j = SP_k^j \times KD^j$. As the standard deviation of MSK_k may reflect the evolutionary variance of the residue k along with hydrophobicity, we use it as the kth element of vector V_i. More details of this vector representation can be found in our previous work [12].

For the residue i, therefore, it is represented by a 1×19 vector V_i; the corresponding target value T_i is 1 or 0, denoting whether the residue is located in an interface or a non-interface region. Our model aims to learn the mapping from the input vectors V onto the corresponding target array T. Our model is trained to make its output as close to the target T as possible.

2.3 Three measures integrated for outlier detection

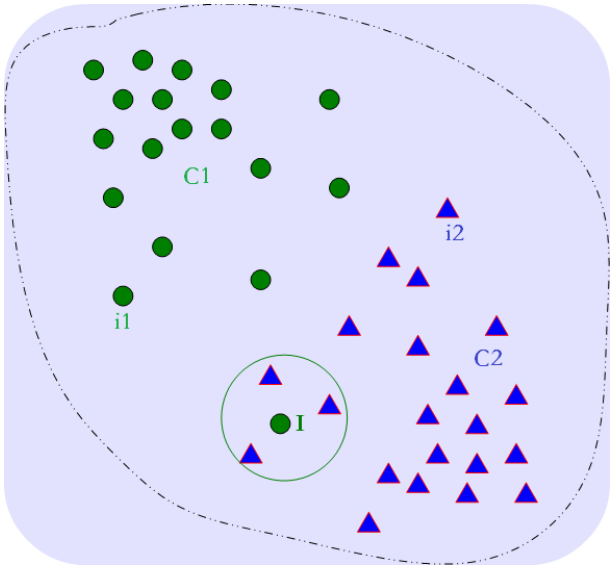


Fig. 1. Example of detecting class outliers. C_1 and C_2 denote the two classes, and i_1 and i_2 are instances representing for class 1 (green points) and class 2 (blue points), respectively.

2.3.1 K-distance of a residue

For a positive integer K, the K-distance of a residue instance I is the mean distance between the instance I and its K nearest neighbors. It describes how far the K nearest instances are away from I on average.

$$K_{dist}(I) = \frac{1}{K}\sum_{m=1}^{K} d(I, i_m) \quad (1)$$

where $d(*,*)$ is the Euclidean distance measurement, and i_m is one of the K nearest neighbors of I.

2.3.2 Probability of the Class Label (PCL) of a residue

The second measure, PCL, is related to the probability of the class label of an instance in terms of its K_{NN} nearest neighbors. For example, the PCL of the instance I (the green one within the circle in Figure 1), denoted by $PCL(I)$, is defined as the ratio of the number of instances with class label 1 to the total number of instances in the green circle in terms of its K_{NN} nearest neighbors, including the instance itself. Therefore, $PCL(I) = 1/4$ if $K_{NN} = 4$ for instance I in Figure 1.

2.3.3 Importance of Within-class and Between-class (IWB)

IWB measures the importance of within-class and between-class changes for an instance. The IWB of instance I, denoted by $IWB(I)$, is defined as the change in the ratio of between-class scatter S_b to within-class scatter S_w before and after excluding instance I. In the two-class case, S_b stands for the subtraction of the mean values of the classes from each other. In contrast, S_w denotes the summation of the two scatters calculated within the same class. In general, a within-class scatter is equivalent to the variance in the same class computed as $S_j = (\sum_{i_1 \in C_j}(i_1 - m_1)^2)^{1/2}$, where $j = 1$ or 2, and C_j is the residue set of class j. In particular, $\widetilde{S_b}$ and $\widetilde{S_w}$ denote between-class scatter and within-class scatter after excluding the instance I, respectively.

$$IWB(I) = \frac{S_b}{S_w} - \frac{\widetilde{S_b}}{\widetilde{S_w}} = \frac{|m_1 - m_2|^2}{S_1 + S_2} - \frac{|\widetilde{m_1} - \widetilde{m_2}|^2}{\widetilde{S_1} + \widetilde{S_2}}$$
$$= \frac{|m_1 - m_2|^2}{(\sum_{i_1 \in C_1}(i_1 - m_1)^2)^{1/2} + (\sum_{i_2 \in C_2}(i_2 - m_2)^2)^{1/2}}$$
$$- \frac{|\widetilde{m_1} - \widetilde{m_2}|^2}{(\sum_{i_1 \in C_1}(i_1 - \widetilde{m_1})^2)^{1/2} + (\sum_{i_2 \in C_2}(i_2 - \widetilde{m_2})^2)^{1/2}} \quad (2)$$

where m_1 and m_2 are sample means for particular classes, and $\widetilde{m_1}$ and $\widetilde{m_2}$ are sample means for the two corresponding classes excluding the instance I.

2.3.4 Class Outlier Score (COS)

The class outlier score of an instance I stands for the degree of an instance being outlier with respect to a particular class.

$$COS(I) = \alpha * PCL(I) + \beta * K_{dist}(I) + \gamma * IWB(I) \quad (3)$$

where α, β, and γ are parameters to trade off the probability of class label, K-distance, and IWB, respectively. In this work, α, β, and γ are all normalized in the range [-1,1]. The outlier detection was performed by a grid search [9]. An instance I is assigned as an outlier residue if

$$COS(I) \geq c \quad (4)$$

where c is a threshold according to experimental results.

From the definitions above, the larger the values of the three measures, the more likely the instance I could be an outlier. Residues with the top scores are treated as outliers, where the exact number of outliers depends on a specific data set.

Since each set of parameters makes different results of the protein-protein interface (PPI) prediction, we aim to find the optimal set of parameters. The most common and reliable approach to parameter selection is to determine parameter ranges, and then to conduct an exhaustive grid search over the parameter space to find the best setting, which is what we have done in this work. A implicit reason is in that the performance is varied in terms of the set of parameters.

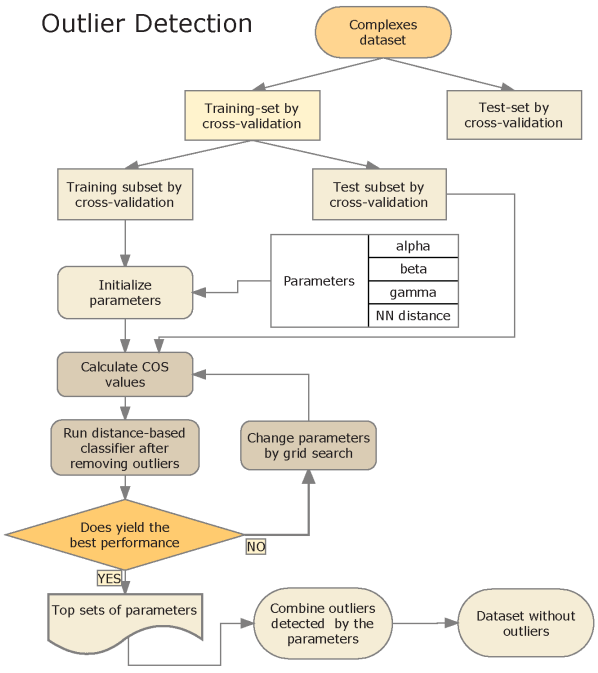


Fig. 2. Flowchart of outlier detection in the training data. The test set is used for evaluating interface prediction of the whole method after outlier detection.

2.3.5 *Outlier detection in the training data*

As shown in Figure 2, our protein complex data set is divided into five subsets for a 5-fold cross-validation to evaluate our proposed method. Our method repeats five times with different training set and test set, in each time one subset is taken as a test set S_{test} and the others are as training set $S_{training}$. Overall performance on our whole data set is yielded by averaging the five experiments.

In each time, in order for detecting outliers in the training data, the training set is separated as a training subset $S^1_{training}$ and a test subset $S^2_{training}$ by another 5-fold cross-validation only in the training step. After parameters such as α, β, γ, and nearest neighbor distance are initialized, the class outlier score ($COS(I)$) is calculated for each residue in $S^1_{training}$, followed by outlier identification based on the outlier detection rule (Eq. 4). Then, for every residue in the test subset $S^2_{training}$, a simple distance-based Euclidean classifier is applied after removing the outliers in $S^1_{training}$. The label of the test residue is predicted as that of the nearest class center. If the classifier does not yield the best performance, then the parameters are changed by a grid search, and the outlier detection and classification for the test subset residues are repeated. If several top sets of parameters give rise to the best classification performance, then a residue is finally identified as an outlier if it is detected as outlier under all the sets of parameters. Then the training data set without these finally confirmed outliers is taken as input into a classifier, e.g. SVM, for learning, in order to identify whether residues, in the pre-reserved test set S_{test}, are interacting or not.

2.4 SVM-based classifiers

Interface residues (positive samples), amounting to only 27.56% in the entire collection of samples, is much less than non-interface residues (negative samples). This leads to a rather unbalanced data distribution. To overcome this problem, the training positive and negative samples are divided into several non-overlapping subsets having roughly the same maximum size. In this work, the positive training samples are split into 2 (M) subsets each consisting of 71205 samples, and the negative training samples are divided into 5 (N) parts each consisting of 74860 samples. Then, we pair these positive and negative training subsets to produce 10 training data sets (namely, 10 $S_{training}$) as illustrated in Figure 3.

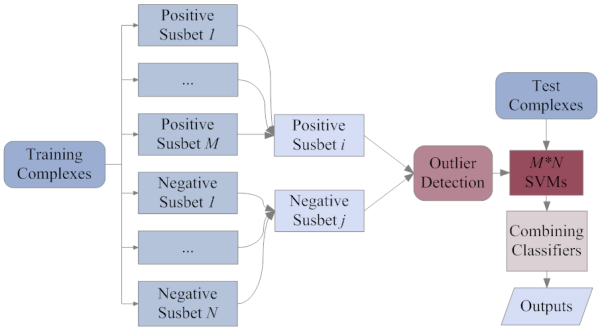


Fig. 3. Flowchart of the interface prediction. In this work, $M = 2$ and $N = 5$ for the construction of SVM ensemble.

SVM is a state-of-the-art classification technique which enjoys excellent generalization performance [48]. The SVM learner is built to judge whether a residue is located at an interface region or not. In this work, a support vector machine with radial basis kernel was applied to identify protein interface residues. The parameters for SVM are set as: $Gamma = 1$ and $Cost = 1$.

The output of the SVMs are combined in a simple way in this paper. A residue is labeled as interacting if half of SVMs identify it as positive class 1; otherwise, it is identified as non-interface residue. In Figure 3 there are $M \times N$ number of SVM classifiers after removing outliers for each pair of balanced training positive and negative subsets i and j. As described, M and N are set to 2 and 5 in this work, respectively, that is, there are 10 sub-classifiers for the SVM ensemble.

In summary, our protein interface prediction by outlier detection and ensemble SVM learning consists of the following five major steps:

- Step 1: Construct balanced training and test subsets $S_{training}$ and S_{test}, i.e., determine the biggest N and M.

- Step 2: Detect interface outliers and non-interface outliers in each training subset $S_{training}$ by a grid search with parameters α, β, γ ranging from -1 to 1 with step size of 0.2, and nearest neighbor (NN) distance ranging from 1 to 20 with step size of 2.
- Step 3: Rank F1 scores on $S^2_{training}$ yielded by the all possible combinations of these parameters, the larger the F1 score, the more effective the outlier detection is.
- Step 4: Confirm outliers for each training subset $S_{training}$, and an SVM is learned by using the training data without the outliers.
- Step 5: Integrate the output of all of the $N \times M$ SVMs to yield the final prediction for each residue in the pre-reserved test set S_{test}.

Our method is named PPI-OD (prediction of protein interface by outlier detection).

2.5 Performance evaluation measures

Since the data set contains imbalanced positive samples and negative samples, only 27.56% negative ones in the data set, in this work six evaluation measures are used to show the performance of our model: sensitivity (Sen), specificity (Spec), accuracy (Acc), precision (Prec), F-measure (F1), and Matthews correlation coefficient (MCC). Their definitions are as follows:

$$
\begin{aligned}
Sen &= \frac{TP}{TP+FN} \\
Prec &= \frac{TP}{TP+FP} \\
F1 &= 2 \times \frac{Prec \times Sen}{Prec + Sen} \\
Acc &= \frac{TN+TP}{TN+FP+FN+TP} \\
Spe &= \frac{TN}{FP+TN} \\
MCC &= \\
&\frac{TP \times TN - FP \times FN}{\sqrt{(TP+FN)(TP+FP)(TN+FP)(TN+FN)}},
\end{aligned} \tag{5}
$$

where TP (True Positive) is the number of interface residues; FP (False Positive) is the number of false positives; TN (True Negative) is the number of non-interface residues; and FN (False Negative) is the number false negatives. In this work, MCC ranges from -1 to 1 and all others are represented by percentage values.

Moreover, the area under the receiver operating characteristic (ROC) curve (AUC) is considered as a performance measure for machine learning algorithms. The definition of AUC is from literature [7], which is calculated using trapezoidal integration.

3 Results and Analysis

3.1 Improved Interface Residue Prediction by PPI-OD

As introduced, the outlier detection step contains a grid search to optimize the parameters α, β, and γ. In this work, top 100 sets of parameters in outlier detection are used to obtain preliminary outliers which are subsequently selected by a majority voting to determine the final outliers. Training data sets without the final outliers are then input to the SVM as proposed above. The overall prediction performance is shown in Figure 4. We can see that our method yields the best performance crossing at the vertical line in Figure 4, where an MCC of 0.55 and an accuracy of 83.12% are achieved. Prediction performance without outlier detection is also listed in Table 1. The PPI-OD method achieves an MCC improvement of around 0.06 and an F1 improvement of around 3.6% in comparison to the method without the outlier detection. Moreover, compared to the model without outlier detection, the model of PPI-OD yields improvements of 2.35% in Sen, 1.41% in Spe, 1.67% in Acc, and 6.56% in Prec. In these experiments, about 5.8% of the training residues are detected as outliers. In order to compare with random prediction, a random predictor is built based our data set and it runs 100 times. By averaging those results, the random predictor has a sensitivity of 72.4%, a precision of 27.56%, and an F1 of 39.93%. It can be seen that our PPI-OD yields significant improvements on the six measures compared to a random predictor.

TABLE 1
Comparison between methods with and without outlier detection.

Method	Sen	Spe	Acc	MCC	Prec	F1
PPI-OD	45.55	97.41	83.12	0.55[§]	86.98	59.79
SVM[‡]	43.20	96.00	81.45	0.49	80.42	56.21

[‡] Method without the use of outlier detection.
[§] In the paper, except for using decimal for the measure MCC, other measures present by percentage, if not specified.

At the step of outlier detection, we tried two classifiers: the Euclidean classifier and a Liblinear classifier [28]. The distance-based Euclidean classifier makes use of the distance that the residues are away from the two class centers, and it assigns the nearest class center label to a residue. Liblinear classifier is a linear classifier for solving large-scale regularized linear classification and it is very efficient on large sparse data sets. It supports logistic regression and linear support vector machines. Experimental results show that the Euclidean classifier outperforms the Liblinear classifier by about 0.02 in MCC (see Figure 5) if MCC and F1 are taken as directly comparable measures for the results by the other methods. It is noted that the Liblinear classifier outperforms the

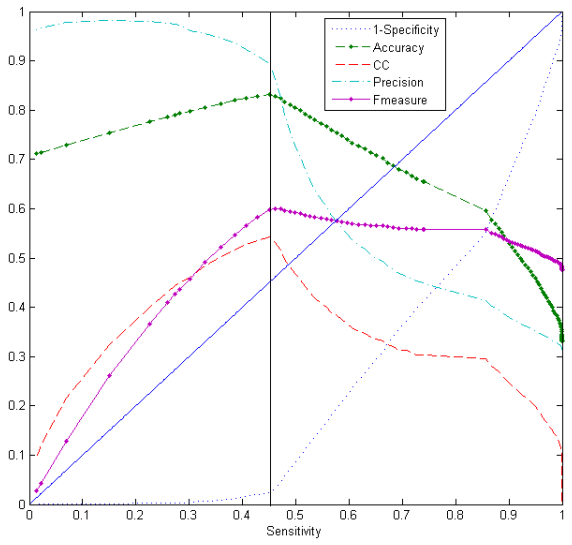


Fig. 4. Prediction performance by PPI-OD.

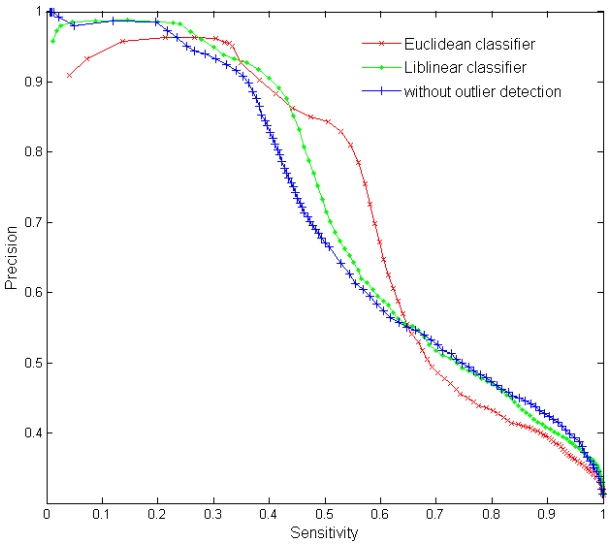


Fig. 5. Prediction performance comparison under three different experiments.

Euclidean classifier by about 0.026 if AUC is used as a comparable measure. In detail the Liblinear classifier, Euclidean classifier, and the model without outlier detection achieve AUCs of 71.05%, 68.44%, and 65.13%, respectively. In this work, MCC and F1 are used as directly comparable measures for the results by other prediction methods.

As expected, the ensemble SVM classifier outperforms the individual classifiers. Table 2 shows the performance comparison between the individual SVMs and the ensemble SVM. It can be seen that the ensemble approach outperforms the best individual SVM by about 0.04 in MCC and 3.3% in F1 measure. Other measures, such as Sen, Acc, etc., all get improvements by the use of ensemble approach as well. Moreover from Table 2, the individual SVM incorporating outlier detection yields a better performance than that without outlier detection by about 0.08 in MCC and 8% in F1 measure. Similarly, other measures get improvement by the use of ensemble approach in the model without outlier detection. In addition, these individual SVMs have similar performance probably due to the similar training data distribution.

PPI-OD costs much extra time compared to the approach without using outlier detection. However, after outliers are detected, the prediction by our model is much simpler than the literature methods. It is due to the simpler encoder of input vectors for the classification problem where input vector in our method is just 19 dimensions, which is less than data used by the past methods, such as the data with 180 dimensions in [46] and the data with 1050 dimensions in [11]. Moreover, the method using the outlier detection outperforms that without it, as discussed in Table 2. Therefore, it is worth of taking the cost of outlier detection as an effective preprocess for interface prediction.

TABLE 2
Prediction performance by the individual models of PPI-OD and by those without outlier detection, and by the ensemble approach.

Method	Sen	Spe	Acc	MCC	Prec	F1
model 1	42.09	97.09	81.94	0.51	84.64	56.23
model 2	42.26	97.16	82.03	0.51	84.99	56.45
model 3	42.22	97.14	82.01	0.51	84.89	56.40
model 4	42.34	97.20	82.08	**0.51**	85.18	**56.57**
model 5	42.14	97.12	81.97	0.51	84.78	56.30
model 6	42.17	97.13	81.99	0.51	84.83	56.34
model 7	42.16	97.13	81.98	0.51	84.81	56.33
model 8	42.08	97.09	81.93	0.51	84.63	56.21
model 9	42.04	97.07	81.91	0.51	84.54	56.16
model 10$^{\perp}$	41.98	97.05	81.88	0.51	84.43	56.08
Ensemble	45.55	97.41	83.12	0.55	86.98	59.79
model 1$^{\top}$	35.00	96.37	79.46	0.43	78.57	48.43
model 2	34.86	96.31	79.38	0.42	78.24	48.23
model 3	33.04	97.00	79.37	0.42	80.73	46.89
model 4	35.11	96.41	79.51	0.43	78.81	48.58
model 5	35.14	96.42	79.53	**0.43**	78.87	**48.62**
model 6	35.01	96.37	79.46	0.43	78.59	48.44
model 7	34.87	96.32	79.38	0.42	78.27	48.25
model 8	35.05	96.38	79.48	0.43	78.67	48.50
model 9	35.13	96.42	79.53	0.43	78.88	48.61
model 10	34.98	96.36	79.44	0.43	78.52	48.40
Ensemble	43.20	96.00	81.45	0.49	80.42	56.21

$^{\perp}$ The ten models above are of PPI-OD.
$^{\top}$ The ten models below do not use outlier detection.

3.2 DPX and CX analysis on outliers and non-outliers

To show the effectiveness of our PPI-OD method, two indices of residues, DPX [41] and CX [40], are

adopted to visualize proportions of outliers and non-outliers. Figure 6 shows the residue composition with respect to DPX, which calculates a depth index for the buried atoms, and makes it possible to analyze the distribution of buried residues. The larger the DPX for one residue, the more likely it could be buried in the protein interior in natural structure and, the more possible it is assigned as non-interface residue. However, atoms that are buried near the protein surface might be involved in interactions with other molecules [41]. We used the PSAIA software in computing DPX for each residue in a complex [36]. In the lower right subgraph, there are more residues having a low DPX value and less residues having a high DPX value in complexes. Our PPI-OD can distinguish more non-interface residues having a low DPX value as outliers (red oval in the upper right subgraph). For interface residues, a similar phenomenon can be found. More residues having a high DPX value are distinguished as outliers (red oval in the upper left subgraph). In addition, residue GLY appears the most possible in proteins interior and residue LEU is more likely located on the interface regions.

Figure 7 illustrates the residue composition with respect to CX. CX computes an atomic protrusion index that makes it possible to highlight the protruding atoms within a protein 3D structure. The larger the CX for one residue, the more likely that it is located on the surface of the protein and the more likely it is interacting with water or residues in a partner protein. Moreover, most of the experimental cleavage sites correspond to residues having a high CX value at the C atom [40]. However, interface residues having a relatively lower CX value might influence the distinguishability of interface predictor. The PSAIA software [36] was used to compute the CX for each residue in a complex. In the lower right subgraph, there are more residues having a low CX value and less residues having a high CX value in complexes while, in the upper left subgraph, more residues have a high CX value. Our PPI-OD can distinguish more interface residues having a low CX value as outliers (red oval in the upper left subgraph). It is difficult to recognize more non-interface residues having a high CX value as outliers. In addition, residues GLY and ALA occur more often in proteins' interior and residues ARG and LYS are more likely to be located at interface regions.

Interestingly, hydrophobic residues on non-interface regions have a lower CX value while hydrophilic residues on interface regions have a higher CX value. However, it is not always the case for the DPX distributions. Unlike the case discussed in Figure 6, hydrophobic residues on the interface regions seem to have a higher DPX value while residues S, T, G, A – having a lower DPX value – appear equally on the interface and non-interface regions.

4 Overall comparison to other methods and performance improvement

We carried out a direct comparison with those methods discussed in [50] on two benchmark data sets: the CAPRI targets(http://capri.ebi.ac.uk/) and the Enz35 subset from the Docking Benchmark 2.0 [37]. As done by previous work [50], only unbound structures were chosen for interface prediction. The Enz35 data set consists of 35 proteins after filtering at 35% sequence identity. The real interface residues are defined as those with a cross-interface contact of $< 5\mathring{A}$ in the native complex. Figure 8 and Figure 9 show the comparison between the six web servers and our method. The six web servers are: PPI-Pred, which takes six protein properties (including surface shape and electrostatic potential) as input [6]; SPPIDER, which is a neural-network method taking predicted solvent accessibility as input [42]; cons-PPISP, which is also a neural-network method but uses PSI-Blast sequence profile and solvent accessibility as input [10]; Promate, which is a naive Bayesian method based on secondary structure, atom distribution, amino-acid pairing and sequence conservation [38]; PINUP, which is an empirical scoring method consisting of side-chain energy, solvent accessible area, and sequence conservation conservation [34]; and Meta-PPISP, whose raw scores are closely related to cons-PPISP, Promate and PINUP by a regression [50].

Figure 8 shows the performance on the CAPRI data set. At the precision level of 50%, the sensitivity of our method PPI-OD is 38%, while the best of the six web servers is just 31%. The worst precision level by PPI-OD is near 20%, while the worst precisions for the six web servers are all below 10%. In the case of precisions above 90%, our model achieves sensitivities slightly below 20% while the six web servers have sensitivities around 15%. In the case of sensitivities from 20% to 90%, our model achieves higher precisions than all of the six literature methods.

Figure 9 shows the performance comparison on the Enz35 data set. At the precision level of 50%, the sensitivity of our method is 67%, while the best of the six web servers is only 50%. In the case of sensitivities below 30%, our model achieves precisions higher than 90%. Fore sensitivities from 30% to 70%, our model also achieves higher precisions than the six methods. Furthermore, The AUC for sensitivity-precision of PPI-OD is the largest among the seven methods. Our method achieves AUCs of 70.12% and 69.42% on CAPRI and Enz35, respectively. The details are shown in Table 3. Despite the small size of the CAPRI and Enz35 data sets, these results are suggestive of the promising prediction capability by our method.

Table 4 shows an indirect comparison with Sikic's methods [46] as it uses a data set overlapping only partly with our data set. Sikic's methods were eval-

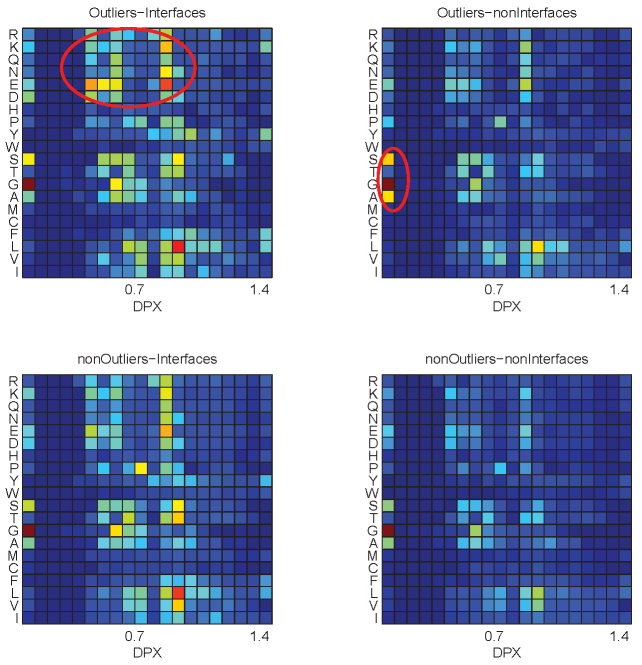


Fig. 6. Residue composition for outliers and non-outliers. Here, y axis stands for residues sorted by hydrophobicity in terms of Kyte measurement [31]. The last column is for all residues having a DPX value more than 1.4. Residue R is more hydrophilic than residue I. Residues are labeled with one-letter codes. In addition, the more bright the small square shows, the more frequent the corresponding residue pair occurs in interface or non-interface region.

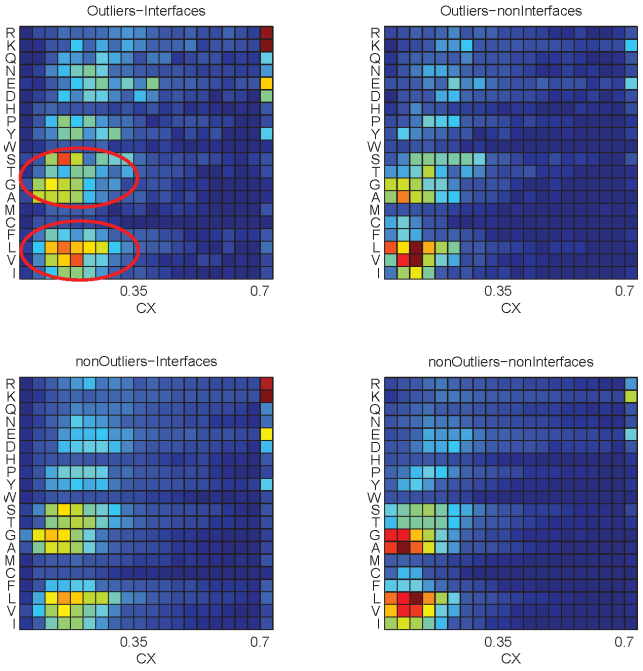


Fig. 7. Residue composition for outliers and non-outliers. Here, y axis stands for residues sorted by hydrophobicity in terms of Kyte measurement [31]. The last column is for all residues having a CX value more than 0.7. Residue R is more hydrophilic than residue I. Residues are labeled with one-letter codes. In addition, the more bright the small square shows, the more frequent the corresponding residue pair occurs in interface or non-interface region.

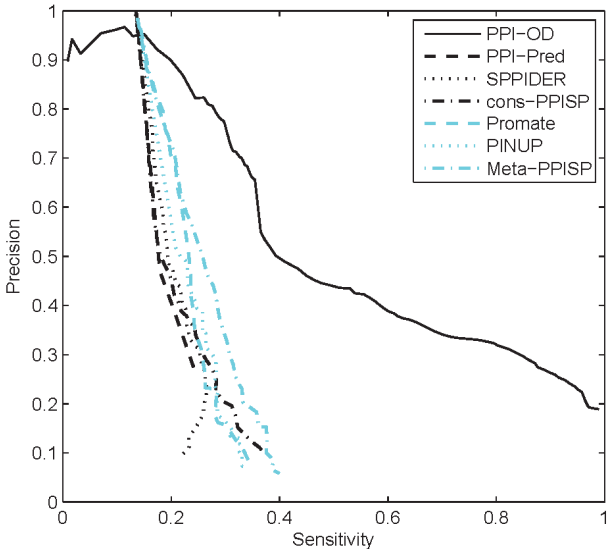


Fig. 8. Performance comparison between the six web servers and PPI-OD on CAPRI.

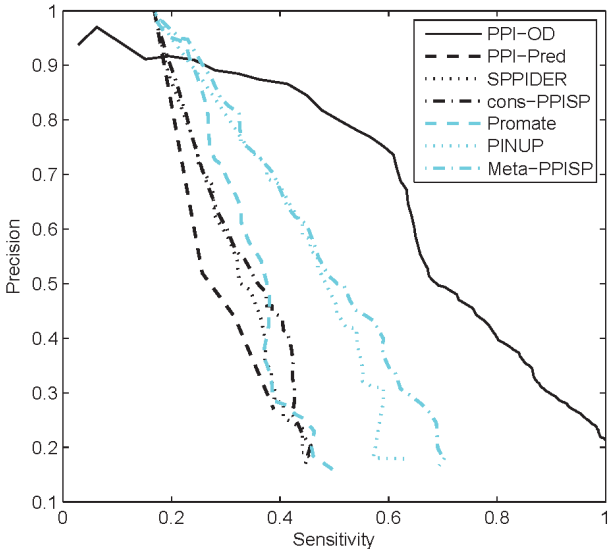


Fig. 9. Performance comparison among the six web servers and PPI-OD on Enz35.

TABLE 3
AUC comparison among the six web servers and PPI-OD on Enz35 and CAPRI.

	PPI-OD	PPI-Pred	SPPIDER	cons-PPISP	Promate	PINUP	Meta-PPISP
CAPRI	70.12	27.98	42.27	47.24	45.80	43.58	46.37
Enz35	69.42	32.34	44.74	44.18	46.38	47.46	47.78

uated on a similar large-scale data set as ours. In addition, the interface ratio of the residues used by Sikic's method is roughly the same as ours as well. As seen in this table, our PPI-OD method performs better than Sikic's. Our method yields a higher sensitivity, precision, and therefore a much higher F1 than Sikic's methods. Moreover, it should be noted that the second method by Sikic is based on the real secondary structure information. Using real structure information may lead to improve the interface prediction indeed.

TABLE 4
Comparison with Sikic's methods.

Method	Type	Ratio	Sen	Prec	F1
PPI-OD	SVM	0.28	45.01	89.35	59.86
Sikic *et al.*	Random Forests	0.27	26.42	84.43	40.25
Sikic *et al.*§	Random Forests	0.27	38.06	76.45	50.82

§ Trained with secondary structure information.

TABLE 5
Interface residues of our method and Sikic's method with different interface definitions.

		Interface residues	
		Non-in common	In common
1C1YA	PPI-OD	K31 F64 M67	E3 V21 V24-Q25 I27 V29 D33 P34 I36-R41 M52 E54 L56 T71
	Sikic *et al.*	K42 Q63	
1C1YB	PPI-OD		T57 R59 N64-V69 N71 K84 K87-G90
	Sikic *et al.*	S55 F61 V70 R73 A85 L91	

§ Residues here are labeled as one letter followed by the number in chain.

An indirect comparison between our method and Sikic's methods [46] is conducted by a case study on the complex PDB:1C1Y. Sikic et al. use a different definition of interface residues, in which a residue was defined to be involved in a protein-protein interaction if any of its atoms were within 6 Å of any atom in a neighboring non homologous chain. Table 5 shows true interface residues of our method and Sikic's method with different interface definition. From Table 5, there are 35 and 40 interface residues for our model and Sikic's model, respectively. In addition, there are 32 interface residues in common, and it should be noted that interface residues non-in common are neighboring each other or located around those interface residues in common in three-dimensional structure of complex. In this case, interface residues defined by the two ways, ASA change with threshold 1 Å and distance between residues in different chains with threshold 6 Å, have approximate ratio of interface residues to non-interface residues. Without using outlier detection, our method achieves a sensitivity of 100% and a precision of 61% (Figure 10(b)); whereas if using outlier detection, a sensitivity of 100% and a precision of 73% are obtained (Figure 10(c)). Under this example, our predictions covers more number of interface residues and the precision is higher than Sikic's (Figure 10(a)). More importantly, our method can correct those false predictions by Sikic or by the method without using outlier detection, e.g., for A118 in chain B. For the wrong predictions by our method, some residues are very close to the partner chain, such as residues G12, T61 in chain A. (All of these figures are plotted by PyMOL, version 1.3 [45].)

TABLE 6
True interface residues in PDB:1SYX.

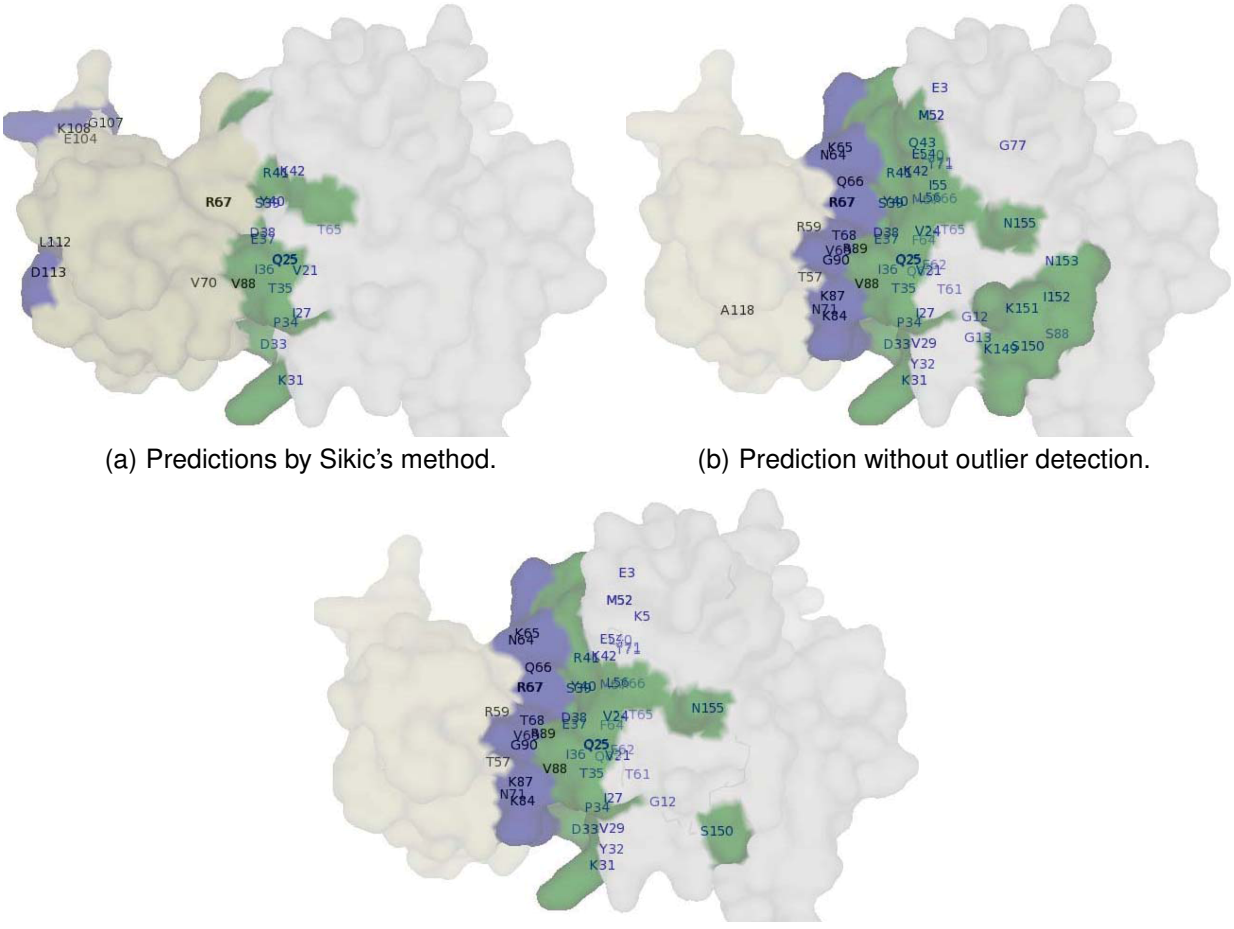
Chain A	K51	D93-G95	D108	Q110-E111
	D114	E117-T118	R121-G122	K125
	R127	V130-D135		
Chain B	V26	E29	E39-G42	R65-Q73
	Y75-N76	R79-D81	D83-T86	

§ Residues here are labeled as one letter followed by the number in chain.

Another case study is carried out to highlight again the performance difference when the outlier detection idea is not used. This case study is performed on the complex 1SYX of a spliceosomal U5 snRNP-specific 15 kDa protein binding to a CD2 antigen cytoplasmic tail-binding protein 2. For the method without using outlier detection, it has a sensitivity of only 25% while most of the predicted interface residues are correct (Figure 11(a)). When applying the PPI-OD method, the sensitivity is increased to 57%, and almost all of the predicted interface residues are also correct (Figure 11(b)). Therefore, the outlier detection idea is effective to predict more number of correct interface residues. Nevertheless, a few predicted residues are still wrong, such as G42 in chain B. The true interface residues in chain A and chain B are listed in Table 6. In summary, there are 7 out of 24 residues are true positive and 17 residues are missed out in chain B. Similarly, 5 residues are true positive and 14 true positive residues are missed out in chain A. Moreover, no predicted residues are false positive in chain A or chain B.

5 Conclusions

In this work, we adopted three outlier detection measures to evaluate the extent a residue becomes an outlier. Results show that the three measures are effective to detect outliers in the training data and make the identification of interface residues more accurate. Our method PPI-OD achieves an MCC improvement of about 0.06 and an F1 improvement of about 3.6% compared to the method without using outlier detection. Compared to past methods that predict protein interface residues, such as Sikic's method [46] and Meta-PPISP [50], our model performs better on a data set with similar interface ratio. Our approach can detect more non-interface residues having a low DPX

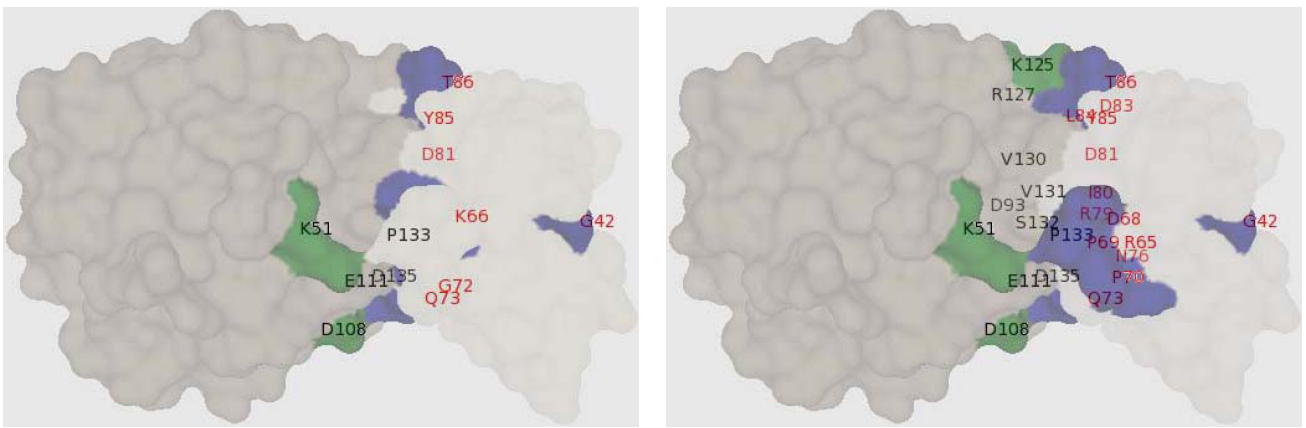


(a) Predictions by Sikic's method.

(b) Prediction without outlier detection.

(c) Prediction by our PPI-OD method.

Fig. 10. Prediction and comparison on complex 1C1Y. The left part stands for chain B (colored in wheat) and the right one is for chain A (colored in gray). The predicted interface residues in chain B are in blue. The predicted interface residues in chain A are in green. Residues here are labeled as one letter followed by the number in chain. The raw data in (a) is from Sikic's paper [46].



(a) Prediction without outlier detection.

(b) Prediction of PPI-OD.

Fig. 11. Case study on complex 1SYX. The left part stands for chain A (colored in wheat) and the right one is for chain B (colored in yelloworange). The predicted interface residues in chain A are in green. The predicted interface residues in chain B are blue. Residues here are labeled as one letter followed by the number in chain.

value as outliers, and can distinguish more residues having a high DPX value as outliers as well. Moreover, it can distinguish interface residues having a low CX value as outliers.

PPI-OD has advantages over many other interface predictors. First, a residue in our work is represented as a *1-by-19* vector by using a sliding window of length 19. This dimensionality is much smaller than most of the other methods. Therefore, our model is simple and space efficient. More importantly, earlier works reported that using larger number of features for input vectors does not always lead to an improved performance [3]. A machine learning algorithm adopting a simple yet more discriminative representation of a sequence space could be much more powerful and effective than using the original data containing all details [3]. Second, the vector for each residue contains evolutionary context with hydrophobicity. Only two features, namely sequence profile and hydrophobicity for residues used in this work, make our model simpler. Actually, biological properties which may be responsible for protein-protein interactions are not fully understood. Therefore, how to find feasible features or feature transformations in protein interaction prediction remains a challenging problem. Finally, unbalanced data between interface and non-interface residues is also a very challenging issue, which always causes a classifier over-fitting. Results in this paper do indicate that developing a classifier ensemble may be a feasible pathway to deal with this unbalance nature.

ACKNOWLEDGMENT

The authors thank Assistant Professor Steven C.H. Hoi at School of Computer Engineering in Nanyang Technological University for some helpful suggestions and a critical reading of the original version that improved the presentation of the paper.

This research work was supported in part by t-wo Singapore Ministry Of Education Tier-2 grants, T208B2203 and MOE2009-T2-2-004. This work was also supported in part by the National Science Foundation of China (No. 60803107).

REFERENCES

[1] Bahadur, R.P., Chakrabarti, P., Rodier, F. and Janin, J. Dissecting subunit interfaces in homodimeric proteins. Proteins 2003; 53:708-719.

[2] Bahadur, R.P., Chakrabarti, P., Rodier, F. and Janin, J. A dissection of specific and non-specific protein-protein interfaces. J Mol Biol 2004; 336:943-955.

[3] Baldi, P. and Brunak, S. Bioinformatics: The machine learning approach. London: The MIT Press; 2000.

[4] Barnett, V. Outliers in Statistical Data. John Wiley; 1994.

[5] Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., et al. The Protein Data Bank. Nucleic Acids Res 2000; 28:235-242.

[6] Bradford, J.R. and Westhead, D.R. Improved prediction of protein-protein binding sites using a support vector machines approach. Bioinformatics 2005; 21:1487-94.

[7] Bradley, A. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 1997; 30:1145-1159.

[8] Chandola, V., Banerjee, A. and Kumar, V. Anomaly detection: A survey. ACM Comput Surv 2009; 41(3):1-58.

[9] Chang, C.C. and Lin, C.J. LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2011; 2(3):27:1-27:27.

[10] Chen, H. and Zhou, H. Prediction of interface residues in protein-protein complexes by a consensus neural network method: test against NMR data. Proteins 2005; 61:21-35.

[11] Chen X.W., Jeong J.C. Sequence-based prediction of protein interaction sites with an integrative method. Bioinformatics 2009; 25(5):585-591.

[12] Chen, P. and Li, J.Y. Sequence-based identification of interface residues by an integrative profile combining hydrophobic and evolutionary information. BMC Bioinformatics 2010; 11:402.

[13] Chung, J., Wang, W. and Bourne, P.E. Exploiting sequence and structure homologs to identify protein-protein binding sites. Proteins 2006; 62:630-40.

[14] Chakrabarti, P. and Janin, J. Dissecting protein-protein recognition sites. Proteins 2002; 47:334-343.

[15] Demirel, M.C., Atilgan, A.R., Jernigan, R. L., Erman, B. and Bahar, I. Identification of kinetically hot residues in proteins. Protein Sci 1998; 7:2522-2532.

[16] Dong, Q., Wang, X., Lin, L., Guan, Y. Exploiting residue-level and profile-level interface propensities for usage in binding sites prediction of proteins. BMC Bioinformatics 2007; 8:147.

[17] Ezkurdia I, Bartoli L, Fariselli P, Casadio R, Valencia A and Tress M.L. Progress and challenges in predicting proteinCprotein interaction sites. Briefings in Bioinformatics 2009; 10(3):233-246.

[18] Fawcett, T. and Provost, F. J. (1999). Activity Monitoring: Noticing Interesting Changes in Behavior. *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 53C62.

[19] Glaser, F., Steinberg, D.M., Vakser, I.A., et al. Residue frequencies and pairing preferences at protein-protein interfaces. Proteins 2001; 43:89-102.

[20] Guharoy, M. and Chakrabarti, P. Conservation and relative importance of residues across protein-protein interfaces. PNAS 2005; 102:15447-52.

[21] Haliloglu, T., Keskin, O., Ma, B., and Nussinov, R. How Similar Are Protein Folding and Protein Binding Nuclei? Examination of Vibrational Motions of Energy Hot Spots and Conserved Residues. Biophysical Journal 2005; 88(3):1552-1559.

[22] Hawkins, D. Identification of Outliers. Chapman and Hall; 1980.

[23] He, Z., Deng, S., and Xu, X. Outlier detection integrating semantic knowledge. Proc of WAIM02 2002; 126-131.

[24] He, Z., Xu, X., Huang, J., and Deng, S. Mining Class Outliers: Concepts, Algorithms and Applications in CRM. Expert Systems with Applications (ESWA'04) 2004; 27(4):681-697.

[25] Japkowicz, N., Myers, C. and Gluck M. A. (1995). A Novelty Detection Approach to Classification. *Proceedings of the 14th International Conference on Artificial Intelligence (IJCAI-95)*, 518C523.

[26] Jones, S. and Thornton, J. M. Principles of proteinprotein interactions. Proc Natl Acad Sci USA 1996; 93:13-20.

[27] Jones, S. and Thornton, J.M. Prediction of protein-protein interaction sites using patch analysis. J Mol Biol 1997; 272:133-143.

[28] Keerthi, S.S., Sundararajan, S., Chang, K.W., Hsieh, C.J. and Lin, C.J.: A sequential dual method for large scale multi-class linear SVMs. KDD 2008; 10(3):233–246.

[29] Kini, R.M. and Evans, H.J. Prediction of potential proteinCprotein interaction sites from amino acid sequence identification of a fibrin polymerization site. FEBS Lett 1996; 385:81-86.

[30] Kleywegt, G.J. and Jonesm T.A. Phi/Psi-chology: Ramachandran revisited. Structure 1996; 15(4):1395C1400.

[31] Kyte, J. and Doolittle, R. A simple method for displaying the hydropathic character of a protein. J Mol Biol 1982; 157:105-132.

[32] Laskowski, R.A. SURFNET: A program for visualizing molecular surfaces, cavities, and intermolecular interactions. J Mol Graph 1995; 13:323-330.

[33] Levy, E.D., Pereira-Leal, J.B., Chothia, C. Teichmann S.A. 3D complex: a structural classification of protein complexes. PLoS Comput Biol 2006; 2(11):e155.
[34] Liang,S. et al. Protein binding site prediction using an empirical scoring function. Nucleic Acids Res 2006; 34:3698C3707.
[35] Marsland, S. (2001) On-Line Novelty Detection Through Self-Organisation, with Application to Inspection Robotics. Ph.D. Thesis, Faculty of Science and Engineering, University of Manchester, UK.
[36] Mihel, J., Sikic, M., Tomic, S., Jeren, B. and Vlahovicek, K. PSAIA-Protein Structure and Interaction Analyzer. BMC Struct Biol 2008; 8:21.
[37] Mintseris J, et al. ProteinCprotein docking benchmark 2.0: an update. Proteins 2005; 60:214-216.
[38] Neuvirth H, Raz R, Schreiber G. ProMate: a structure based prediction program to identify the location of protein-protein binding sites. J Mol Biol 2004; 338:181-99.
[39] Ofran, Y. and Rost, B. ISIS: interaction sites identified from sequence. Bioinformatics 2007; 23:13-6.
[40] Pintar A, Carugo O and Pongor S. CX, an algorithm that identifies protruding atoms in proteins. Bioinformatics 2002; 18(7):980-984.
[41] Pintar A, Carugo O and Pongor S. DPX: for the analysis of the protein core. Bioinformatics 2003; 19(2):313-314.
[42] Porollo, A. and Meller, J. Prediction-based fingerprints of protein-protein interactions. Proteins 2007; 66:630-45.
[43] Rousseeuw, P. and Leroy, A. (1996) Robust Regression and Outlier Detection, 3nd edn, John Wiley and Sons.
[44] Sander, C. and Schneider, R. Database of homology derived protein structures and the structural meaning of sequence alignment. Proteins 1991; 9:56-68.
[45] Schrodinger, L.L.C. The PyMOL Molecular Graphics System. Version 1.3r1, 1991.
[46] Sikic, M., Tomic, S. and Vlahovicek, K. Prediction of ProteinCProtein Interaction Sites in Sequences and 3D Structures by Random Forests. PLoS Comput Biol 2009; 5(1):e1000278.
[47] Singh, R. Xu, J. and Berger, B. Struct2net: integrating structure into protein-protein interaction prediction. Pac Symp Biocomput 2006; 11:403-14.
[48] Cortes, C. and Vapnik, V. Support-Vector Networks. Machine Learning 1995; 20:273-297.
[49] Wang, B., Chen, P., Huang, D.S., Li, J.J., Lok, T.M., et al. Predicting protein interaction sites from residue spatial sequence profile and evolution rate. FEBS Lett 2006; 580:380-384.
[50] Zhou, H. and Qin S. Interaction-site prediction for protein complexes: a critical assessment. Bioinformatics 2007; 23(17):2203-2209.

Peng Chen is currently an Associate Professor in the Institute of Intelligent Machines, Chinese Academy of Sciences, and in University of Science and Technology of China (USTC), Hefei, P.R.China. He received his Bachelor and Master degrees in Control Science and Engineering from Electronic Engineering Institute and Kunming University of Science and Technology, respectively, and Ph.D degree in Control Science and Engineering from University of Science and Technology of China. From Jan. 2006 to Jun. 2006, he was a senior research associate in City University of HongKong. During Apr. 2008-Apr. 2009, He was a postdoctoral research fellow in Howard University, USA. From Jul. 2009 to Dec. 2010, he worked at Bioinformatics Research Centre, Nanyang Technological University, Singapore, as a postdoctoral research fellow. Dr. Chen's research interests include machine learning and data mining with applications to pattern recognition, bioinformatics, etc. He has published more than 20 high quality referred papers in top conferences and journals, including BMC Bioinformatics, Amino Acids, and FEBS Letters, etc. Dr. Chen has been often invited as PC member and/or referee/reviewer for many premier international conferences and journals, including TNN, NN, IJDMB, DMIN2011, BIFE2010, and others.

Jinyan Li obtained his bachelor degree of science (applied mathematics) from National University of Defense Technology (China), his master degree of engineering (computer engineering) from Hebei University of Technology (China), and his PhD degree (computer science) from the University of Melbourne (Australia). He joined UTS in March of 2011 after ten years of fascinating research and teaching work in Singapore (Institute for Infocomm Research, Nanyang Technological University, and National University of Singapore). His return to Australia marks another milestone of his research life as he is operating on a wide spectrum of bioinformatics research topics at the Advanced Analytics Institute.Jinyan loves research on bioinformatics, computational biology, data mining, graph theory, information theory, machine learning, and theoretical biology. He has published 51 journal articles and 60 conference papers, of which 24 journal papers and 32 conference papers are in the ERA ratings of A/A*. These journals include: Machine Learning, Artificial Intelligence (under my name Jin-Yan Li), Data Mining and Knowledge Discovery, IEEE TKDE, Bioinformatics, Nucleic Acids Research and Cancer Cell. Conference papers include those in KDD, ICML, PODS, ICDT, ICDE, ICDM and SDM. In addition, he edited 3 scholarly books, and published 9 book chapters and 4 patents.Jinyan is widely known for his pioneering and theoretical research work on emerging patterns that has spawned numerous follow-up research interests in data mining, machine learning, and bioinformatics and made an enduring contribution to the fields.

Limsoon Wong is a provost's chair professor in the School of Computing and a professor in the Yong Loo Lin School of Medicine at the National University of Singapore. Before that, he was the Deputy Executive Director for Research at A*STAR's Institute for Infocomm Research. He is currently working mostly on knowledge discovery technologies and is especially interested in their application to biomedicine. Prior to that, he has done significant research in database query language theory and finite model theory, as well as significant development work in broad-scale data integration systems. Limsoon has written about 150 research papers, a few of which are among the best cited of their respective fields. In recognition for his contributions to these fields, he has received several awards, the most recent being the 2003 FEER Asian Innovation Gold Award for his work on treatment optimization of childhood leukemias and the 2006 Singapore Youth Award Medal of Commendation for his sustained contributions to science and technology. He serves on the editorial boards of Information Systems (Elsevier), Journal of Bioinformatics and Computational Biology (ICP), Bioinformatics (OUP), IEEE/ACM Transactions on Computational Biology and Bioinformatics, Drug Discovery Today (Elsevier), and Journal of Biomedical Semantics (BMC). He is a scientific advisor to Semantic Discovery Systems (UK), Molecular Connections (India), and CellSafe International (Malaysia). He received his BSc(Eng) in 1988 from Imperial College London and his PhD in 1994 from University of Pennsylvania.