Protein Contact Prediction Based on Protein Sequences *

Dong-Jian Lin[a], Chang-Biau Yang[a†]and Yung-Hsing Peng[b]

[a]Department of Computer Science and Engineering
National Sun Yat-sen University, Kaohsiung, Taiwan
[b]Innovative DigiTech-Enabled Applications & Services Institute
Institute for Information Industry, Kaohsiung, Taiwan

Abstract

The inter-residue contacts in a protein impact the formation of its protein folds, which plays a critical role in building protein structures. In this paper, we propose a classification method to predict the residue-residue contacts of a target protein, and develop a new measurement to evaluate the accuracy of prediction. We compare three prediction tools, which are the support vector machine (SVM), the k-nearest neighbor algorithm (KNN), and the penalized discriminant analysis (PDA), by referring to the self-testing of the training set derived from representative protein chains from PDB (PDB-REPRDB). After that, we apply the best classifier (SVM) to predict a testing set of 173 protein chains derived from previous study. The experimental results show that the accuracy of our prediction achieves 24.84%, 15.68%, and 8.23% for three categories of different contacts , which improves the result of random exploration (5.31%, 3.33%, and 1.12%, respectively).

Keywords: protein, contact prediction, classification

1 Introduction

The amino acid sequence of a protein can be analyzed and used to explore its protein fold and to predict its protein structure. The interaction between amino acids constructs the protein fold and maintains the structural stability of a protein. In other words, the inter-residue *contact* between amino acids is the cement of protein structures that controls most of the biological functions [8]. This shows that the inter-residue contact plays a crucial role in building protein structures. With the contact information in a protein, the tendency of the protein fold can be observed, and then the construction of the 3D protein structure can be achieved more precisely. Therefore, researchers desire to analyze and to predict the interaction between these residues.

Some methods for protein contact prediction have been proposed, and they can be roughly categorized and introduced as follows. Statistical contact prediction [9, 14] uses the concept of correlated mutation and multiple sequence alignment (MSA) to get the contact map of the target protein. Template-based contact predictors [12, 15] use the template information of known protein structures to find the contacts in the target protein. The machine learning method [3, 5, 16] first fetches the training models from the contact information of known structures, and then uses these models to predict the contacts of the target protein. Also, there are some hybrid approaches [7, 10] that utilize the above three methods.

In this paper, we propose a new algorithm for contact prediction over standard amino acids. In our algorithm, we first develop a measurement for three prediction tools, which are the support vector machine (SVM), the k-nearest neighbor algorithm (KNN), and the penalized discriminant analysis (PDA), determining best classifiers for various situations. After that, we utilize those classifiers to achieve better accuracies. The rest of this paper is organized as follows. In Section 2, we introduce some prerequisite knowledge. In

*This research was partially supported by the National Science Council of Taiwan under contract NSC 100-2221-E-242-003. This research was also partially supported by the "Advanced Sensing Platform and Green Energy Application Technology Project" of the Institute for Information Industry which is subsidized by the Ministry of Economy Affairs of the Republic of China.

†Corresponding author. E-mail: cbyang@cse.nsysu.edu.tw (Chang-Biau Yang).

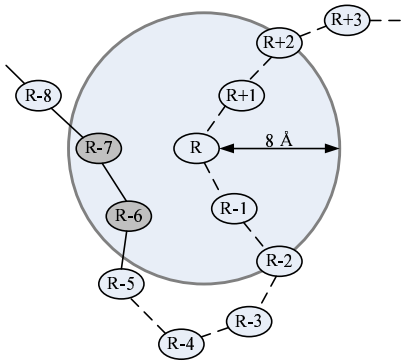

Figure 1: The contacts of residue R [8]. The gray circle denotes the distance threshold 8 Å, and the residues connected with dashed line are not taken into account. The residues R-6 and R-7 are considered in contact with residue R.

Section 3, we describe our approach. The experimental results are given and discussed in Section 4. Finally, in Section 5, we conclude our results and give some future works.

2 Preliminaries

In this section, we first define the residue-residue contact used in our work. Next, we explain some basic knowledge for the position specific scoring matrix (PSSM), the support vector machine (SVM), the k-nearest neighbor algorithm (KNN), and the penalized discriminant analysis (PDA).

2.1 Definition of Contacts

In this paper, two residues in the same protein are said to be in *contact* if the distance between their C_α atoms is less than 8 Å[16]. In addition, the sequence separation between the two residues that form a contact is at least 6, as shown in Figure 1. With this rule, we can divide the contact residue pairs into three categories, which are of short, medium, and long ranges in their sequence separation. The sequence separation between two residues of a short-range contact, a medium-range contact, and a long-range contact are 6 to 11, 12 to 24, and more than 24, respectively [3, 16], as shown in Table 1. At the same time, a residue pair that is not a contact is called a *noncontact*, which can also be categorized with Table 1.

By investigating most protein structures in our data, we find that the ratio of noncontacts to contacts is at least 18:1. To avoid the bias in prediction, we rearrange the ratio of noncontacts to contacts as 1:1, 2:1, 3:1, and 4:1 by randomly selecting noncontacts, generating four feasible training datasets. In our approach, we will use these four newly generated training datasets to construct the prediction mechanism.

Table 1: The three categories of residue pairs.

	Short	Medium	Long
Separation	$6-11$	$12-24$	>24

2.2 Position Specific Scoring Matrix

The *position specific scoring matrix* (PSSM), proposed by Altschul *et al.* [2], is a commonly used scoring matrix for biological sequences. In a PSSM, the first row represents the twenty standard amino acids, and the first and the second column represent the sequence number and the type of all amino acids in the target, respectively. Then, the score in each matrix entry stands for the probability at each position that certain residue may occur. A higher score in a row means that the amino acid at this position may perform the same biologic function in homologous sequences. In order to obtain the PSSM, we invoke the *basic local alignment search tool* (BLAST) and the *position-specific iterated-BLAST* (PSI-BLAST) developed by the National Center for Biotechnology Information (NCBI).

2.3 Support Vector Machine

The *support vector machine* (SVM), proposed by Corinna Cortes *et al.* [6], is a set of machine learning techniques for data classification. Basically, SVM constructs a hyperplane to split a set of samples into two classes. It is widely applied in many research fields. For data classification, SVM first takes a set of input training data that belong to one of the classes, and then extracts feature vectors from the input. After that, SVM produces a training model by some training algorithms. Then, according to this training model, SVM predicts to which class a testing sample belong. Therefore, the training procedure and the predicting procedure are the main functions of SVM in data classification. In this paper, we use LIBSVM [4], a useful software package for SVM implementation, to perform our experiments. LIBSVM accepts the input training

data, automatically generates the training model, and then executes the prediction with the training model.

2.4 The K-nearest Neighbor Algorithm

The *k-nearest neighbor* (KNN) algorithm is a popular machine learning method for classifying objects. With training data set, the KNN algorithm first finds out the k nearest neighbors (in the feature space) for a given testing sample. Then, it takes the testing sample into the class that the most neighbors belong to. To avoid ties, the value of k is usually set to a positive odd. In this paper, we set k to a number greater than 1 for the voting mechanism. In our experiment, we apply the software package Machine Learning PYthon (MLPY) [1] to perform KNN. In our approach, the value of k varies from 3 to 201 for optimization. According to our observation, the prediction accuracy for self testing increases with smaller k and decreases gradually with larger k.

2.5 Penalized Discriminant Analysis

The *penalized discriminant analysis* (PDA) proposed by Hastie *et al.* [11] is a data-analytic tool. Briefly, the concept of PDA is to solve the following equation.

$$y = X\beta, \tag{1}$$

where y stands for an $n \times 1$ response vector (class), X stands for an $n \times p$ predictor matrix (training data), and β stands for a $p \times 1$ vector of unknown regression parameters (prediction model). By solving the equation, we can find out the prediction model β, which can be used for performing classification (computing y for a given X). We use Machine Learning PYthon (MLPY) to implement PDA in our experiment. The input format of PDA is identical to that of KNN.

3 Our Prediction Method

In this section, we propose a method that predicts the residue-residue contacts of a target protein. Our method adopts not only the primary structure but also the position specific scoring matrix (PSSM) as features. We evaluate three classifiers by the accuracy of their predictions over the training data, and then use the best classifier to predict contacts for the testing data.

3.1 Contact Extraction

In this part, we explain how to get the contacts and noncontacts of proteins. First, we obtain protein files (including training proteins and testing proteins) from Protein Data Bank (PDB), and then extract the primary sequences and the three-dimensional coordinates of residues from these files. According to the coordinates of residues, we calculate the pairwise distance of residues in a protein, and determine their class with the restriction mentioned in Section 2.1. Then, we separate the contacts and the noncontacts into three categories including short, medium, and long range. For classification, we set the label of contact to 1 and the label of noncontact to 0. In the following, we describe the feature extraction method for contact residues, by which we can transform the training proteins and the testing proteins into feature vectors.

3.2 Feature Extraction

In this work, we include three set of features, which are the sequence separation, the amino acid composition, and the position specific scoring matrix (PSSM).

3.2.1 The Sequence separation

The sequence separation $SS(i, j)$ stands for the number of residues between R_i and R_j, as shown in Equation 2, where R_i and R_j stand for the ith and the jth amino acid in the protein sequence P, respectively. The sequence separation refers to the in-between information of the residue pairs. For example, the sequence separation of the residue pair (R_6, R_{20}) is 14.

$$SS(i, j) = |i - j|\,, \tag{2}$$

where $1 \leq i, j \leq |P|$ and $|P|$ denotes the length of the given protein sequence.

3.2.2 The Amino Acid Composition

In general, the amino acid composition represents the composition ratio of twenty standard amino acids in the primary sequence (See Equation 3). In this paper, the amino acid composition considers the sequence around the target residue, rather than the entire sequence. For each residue R_i, we set a window of size $2k+1$, and then we can fetch a fragment of length $2k+1$ centered on R_i. After that, we compute the amino acid composition

around R_i by Equation 4. In this paper, we set $k = 7$ and fetch a 15-residue fragment on each R_i.

$$aac_{P,a_i} = N_{P,a_i}/|P|, \tag{3}$$

where $i \in \{1, 2, \cdots, 20\}$, a_i denotes the ith amino acid among the twenty standard amino acids, N_{P,a_i} stands for the number of occurrences of amino acid a_i in the sequence P, and $|P|$ stands for the length of the given protein sequence.

$$aac(F_i, k) = N_{F_i,a_j}/(2k + 1), \tag{4}$$

where F_i is the fragment $R_{i-k}R_{r-k+1}\cdots R_{i+k}$ fetched from R_i, and $j \in \{1, 2, \cdots, 20\}$.

Taking F_i='VLSEGEWQLVLHVWA' with $k = 7$ and R_i='Q' for example, we calculate the number of each standard amino acid appearing in this fragment, and then divide the number by 15. In this example, for a_j='E' we have $N_{F_i,a_j}/15 = 2/15 = 0.1333$ as one of the twenty amino acid composition for F_i. Note that for each contact or noncontact sample, the feature size of amino acid composition is 2×20, since each residue keeps twenty amino acid composition.

3.2.3 The Position Specific Scoring Matrix

In this paper, we take the submatrix $PSSM(P, R_i, 2k + 1)$ from the position specific scoring matrix (PSSM) as one of our features. Similarly, R_i stands for the ith amino acid in the protein sequence P, and $(2k + 1)$ stands for the size of window. That is, $PSSM(P, R_i, 2k + 1)$ denotes the submatrix of F_i derived from the PSSM. In our method, we set $k = 7$ and the size of window is equal to 15. Therefore, the obtained submatrix contains 15×20 elements. For each sample (contact or noncontact), the size of this PSSM feature is $2 \times 15 \times 20 = 600$, because each residue keeps a matrix of size $15 \times 20 = 300$.

3.2.4 Normalization of Features

Normalization is an important step for classification that balances the features. Let M_j and D_j be the mean and the standard deviation of the jth feature values over all training samples, respectively. Then, the normalized value of each feature is defined as $\frac{v_{i,j} - M_i}{D_i}$, where $v_{i,j}$ refers to the jth feature value for the ith sample. Note that both training and testing samples should be normalized with M_j and D_j. By doing so, we reduce the variance of features between training data and testing data. In total, there are 641 features included in one sample.

3.3 Evaluation of Classifiers

In this section, we describe how to choose the best classifier from the prediction tools SVM, KNN, and PDA. Recall that we have four training datasets with ratios on the numbers of noncontacts and contacts, including 1:1, 2:1, 3:1 and 4:1. We first divide each training dataset into three subsets of short, medium, and long range. Then, we perform three classifiers for each subset. Finally, for each subset we select the classifier with the highest accuracy. With four training datasets, we have 12 subsets and 36 classifiers. Eventually, we select 12 classifiers for prediction.

For evaluating classifiers, we build three prediction mechanisms based on the training datasets, carrying out the self-testing, and then find out the classifier with the highest accuracy. Our approach for evaluation is given as follows.

Step 1. Get the training protein files of PDB.

Step 2. Extract the contacts and the noncontacts from the protein files.

Step 3. Construct four different training datasets with ratios 1:1, 2:1, 3:1, and 4:1.

Step 4. Separate each training dataset into three categories (short, medium, and long), and set $i = 1$.

Step 5. Construct the training data (features) corresponding to the ith training dataset.

Step 6. Normalize the features over the training data, and store the means and the standard deviations derived from the training data.

Step 7. Build the classification mechanisms of SVM, KNN, and PDA.

Step 8. Implement the self-testing for three classifiers on the training data, and evaluate the performance with the prediction accuracy to choose one classifier for each category.

Step 9. If $i \neq 4$, increase i by 1 and go to Step 5. Otherwise, output the 12 selected classifiers.

3.4 Prediction of Contacts

With the work of Section 3.3, we can obtain 12 classifiers for prediction. Our algorithm for contact prediction is given as follows.

Step 1. Get the primary sequence of the target protein.

Step 2. Extract the residue pairs of the testing proteins for prediction.

Step 3. Separate residue pairs into three categories (short, medium, and long).

Step 4. Extract features from the residue pairs. Set i to 1.

Step 5. Normalize the features of the target protein with the means and standard deviations of the ith training dataset.

Step 6. Execute the prediction with the selected classifiers (short, medium, and long).

Step 7. Output the predicted results and increase i by 1. Go to Step 5 if $i \leq 4$.

In the predicted results (12 sets), the residue pairs are labeled with '0' or '1'. A residue pair labeled with '1' is a predicted contact, whereas a predicted noncontact is labeled with '0'.

4 Experiment

In this section, we first introduce the dataset of our experiments and the filtering rules on the dataset. Next, we explain how to evaluate the performance of prediction. Finally, we show the experimental results.

4.1 Datasets

We perform independent tests in our work. Therefore, the training dataset and the testing dataset are different. In the following, we describe how we get both datasets, and why we utilize these datasets to perform the experiments.

4.1.1 The Training Dataset

The *representative protein chains from PDB* (PDB-REPRDB) [13] is a filter of proteins. Following the constraints in Table 2, we use PDB-REPRDB to build a non-redundant dataset of 1938 protein chains obtained from the Protein Data Bank (PDB). In our experiment, we treat these protein chains as protein sequences. In the training dataset, the pairwise sequence identities are no more than 10%, and the pairwise root mean square deviation (RMSD) is more than 10 Å. Also,

Table 2: The factor and constraints of representative protein chains from PDB (PDB-REPRDB) for selecting protein chains (sequences).

Factor	Constraint
Resolution	<= 2.5 Å
R-factor	<= 0.2
Chain break	no
Ratio of non-standard amino acid residues	<= 1%
NMR	include
Number of residues	>= 50
RMSD	> 10 Å
Sequence identity	<= 10%

each selected sequence has an X-ray crystal structural resolution better than 2.5 Å, and an R-factor less than 0.2. Furthermore, the ratio of nonstandard amino acids in each sequence is under 1%, and all broken sequences are removed.

To keep a feasible amount of data, we reserve 500 protein sequences as training set based on some constraints. First, we divide the 1938 sequences into 16 groups based on the number of contacts in each sequence (See Table 3). Next, we obtain the number of sequences ns_i in the ith group and calculate the proportion defined as follows:

$$prop_i = \frac{ns_i}{1938},\ 1 \leq i \leq 16. \tag{5}$$

In the ith group, we multiply 500 by $prop_i$ to get the number of reserved sequences nr_i. Finally, we randomly select nr_i sequences from the ith group and obtain total 500 protein sequences as the training data. In this way, the new dataset is likely to hold the distribution of contacts in the original dataset. Note that the above approach reduces the size of both contacts and noncontacts of the training set.

Table 4 indicates the information of the training set of 500 sequences in three categories for short, medium, and long ranges. Since the number of samples in Table 4 is still huge, for each category we randomly select 5000 contacts to reduce the number of samples. For each ratio, we also randomly select a corresponding number of noncontacts.

4.1.2 The Testing Dataset

We perform the contact predictions on the testing dataset of 173 protein sequences obtained from the previous study of Fariselli *et al.*[7]. Table 5

Table 3: The information of 16 groups.

Number of contacts	Number of sequences (ns)	Proportion ($prop$)	Number of reserved sequences (nr)
< 200	38	0.0196	10
$200 - 400$	323	0.1667	83
$400 - 600$	453	0.2337	117
$600 - 800$	352	0.1816	91
$800 - 1000$	304	0.1569	78
$1000 - 1200$	180	0.0929	46
$1200 - 1400$	111	0.0573	29
$1400 - 1600$	67	0.0346	17
$1600 - 1800$	33	0.0170	9
$1800 - 2000$	26	0.0134	7
$2000 - 2200$	19	0.0098	5
$2200 - 2400$	10	0.0052	3
$2400 - 2600$	9	0.0046	2
$2600 - 2800$	5	0.0026	1
$2800 - 3000$	4	0.0021	1
> 3000	4	0.0021	1
	1938		500

Table 4: The information of contacts and noncontacts in the training set of 500 protein sequences. The term 'Ratio' denotes the ratio of the noncontacts and contacts. The term 'Size' denotes the computer file size of training data, including the features.

	Short	Medium	Long
Contact	45105	62529	206889
Noncontact	832576	1777263	25210686
Total	877681	1839792	25417575
Ratio	18:1	28:1	121:1
Size	$>$ 10 GB	$>$ 20 GB	$>$ 300 GB

Table 6: The information of contacts and noncontacts in the 173 testing proteins. The term 'Ratio' denotes the ratio of the noncontacts and contacts. The term 'Size' denotes the computer file size of testing data including the features.

	Short	Medium	Long
Contact	8274	10529	30799
Noncontact	147624	305912	2711620
Total	155898	316441	2742719
Ratio	18:1	29:1	88:1
Size	$>$ 2 GB	$>$ 4 GB	$>$ 30 GB

lists the PDBIDs of the testing dataset, and Table 6 shows the information of the testing dataset of 173 sequences in three categories short, medium, and long ranges. Because the long-range file is too huge to the prediction tools, we divide the long-range file into several subfiles that are acceptable for the prediction. This operation does not affect the result of prediction, because these subfiles preserve the original data.

4.2 Performance Evaluation

In the training procedure, there are three categories of residue pairs (short, medium, and long), four ratios of training sets (1:1, 2:1, 3:1, 4:1), and three classifiers (SVM, KNN, PDA). In our experiment, the accuracy of self-testing for prediction is defines as follows:

$$\frac{N_{cp}(i,j,k)}{N_{train}(i,j,k)} \times 100\%, \quad i \in \{1,2,3\}, j \in \{1,2,3,4\}, k \in \{1,2,3\}, \tag{6}$$

where i, j, and k are the indices of categories, training sets, and classifiers, respectively. In the above formula, $N_{train}(i,j,k)$ and $N_{cp}(i,j,k)$ denote the number of residue pairs and the number of correctly predicted residue pairs (true positive and true negative) in the ith category and jth training set with the kth classifier, respectively.

Let $acc(i,j,k)$ be the accuracy of prediction in the ith category that uses the prediction mechanism developed by the jth training dataset, in which k is a number that limits the number of qualified predictions (ranked by confidence obtained from classifiers). The accuracy of prediction in our experiment is defined as follows:

$$acc(i,j,k) = \frac{N_{corr}(i,j,k)}{N_{pred}(i,j,k)} \times 100\%, \quad i \in \{1,2,3\}, j \in \{1,2,3,4\}, k \in \{1,2,\cdots,10\}, \tag{7}$$

where i, j, and k are the index of categories, the index of training sets, and the limit for qualification, respectively. In the above formula, $N_{pred}(i,j,k)$ and $N_{corr}(i,j,k)$ denote the number of predicted

Table 5: The PDBIDs of the 173 testing protein chains.

1a1iA	1c5aA	1pptA	2hfhA	1bd8A	1exgA	1whiA	1bf8A	1thvA	1isoA
1a1tA	1cfhA	1brfA	2hoaA	1beaA	1hfcA	2fspA	1bjkA	1vinA	1kvuA
1a68A	1ctjA	1scoA	2hqiA	1bfeA	1ifcA	2gdmA	1byqA	1xnbA	1moqA
1a7iA	1cyoA	1spyA	1rofA	1bfgA	1jvrA	2ilkA	1c3dA	1yubA	1svbA
1acpA	1fnaA	1sroA	2sn3A	2lefA	1kpfA	2lfbA	1cdiA	1zinA	1uroA
1ah9A	1hevA	1tbnA	2sxlA	1bkfA	1kteA	2pilA	1cneA	2baaA	1yscA
1ahoA	1hrzA	1tivA	3gatA	1bkrA	1bgfA	2tgiA	1cnvA	2fhaA	2caeA
1aieA	1kbsA	1tleA	3mefA	1br0A	1npkA	2uczA	1csnA	16pkA	2dpgA
1ailA	1mbhA	1tsgA	4mt2A	1bsnA	1pdnC	1makA	1ezmA	1a8eA	2pgdA
1ajjA	1mbjA	1ubiA	5ptiA	1bv1A	1pkpA	3lztA	3chyA	1adsA	3grsA
1aooA	1msiA	1uxdA	1a62A	1bxaA	1poaA	3nulA	1jukA	1ftsA	1arvA
1ap0A	1mzmA	2acyA	1a6gA	1c25A	1putA	5p21A	1kidA	1axnA	
1arkA	1nxbA	2adxA	1aczA	1cewI	1ra9A	7rsaA	1mmlA	1b0mA	
1awdA	1ocpA	2bopA	1asxA	1cfeA	1rcfA	1ad2A	1mrjA	1bg2A	
1awjA	1opdA	2echA	1audA	1cyxA	1rieA	1akzA	1nlsA	1bgpA	
1awoA	1pceA	2fdnA	1ax3A	1dunA	1skzA	1ammA	1ppnA	1bxoA	
1bboA	1plcA	2fn2A	1b10A	1ecaA	1tamA	1aolA	1rgsA	1dlcA	
1bc8C	1pouA	2fowA	1bc4A	1ervA	1vsdA	1ap8A	1rhsA	1irkA	

Table 7: The accuracies of self-testing on the short-range residue pairs over four different ratios.

Ratio	SVM	KNN	PDA
1:1	89.58%	78.90%	70.99%
2:1	92.64%	83.76%	70.30%
3:1	99.68%	86.89%	70.35%
4:1	99.58%	88.56%	70.69%

Table 8: The accuracies of self-testing on the medium-range residue pairs over four different ratios.

Ratio	SVM	KNN	PDA
1:1	91.03%	79.30%	67.50%
2:1	99.89%	84.54%	67.83%
3:1	99.83%	86.14%	67.28%
4:1	99.78%	88.45%	66.73%

contacts and the number of correctly predicted contacts, respectively.

4.3 Experimental Results

Tables 7, 8, and 9 show the performance of three classifiers for the short-range, medium-range, and long-range residue pairs, respectively. In Table 7, SVM outperforms the other two in all training datasets. Hence, we select SVM as the prediction mechanism for short-range contacts. Similarly, as Tables 8 and 9 show, SVM is also selected as our classification tool for medium-range and long-range contacts.

Three categories (short, medium, and long ranges) of contacts are predicted in the testing set. Table 10 shows the results of the short-range cases. In this table, the term 'Top' denotes the limit of

Table 9: The accuracies of self-testing on the long-range residue pairs over four different ratios.

Ratio	SVM	KNN	PDA
1:1	92.08%	77.57%	71.28%
2:1	94.83%	80.46%	68.47%
3:1	86.66%	83.68%	68.01%
4:1	87.75%	85.69%	67.74%

qualified predictions (ranked with confidence provided by classifiers), and N_{pred} denotes the number of predicted contacts. As shown in Table 6, the total number of predicted contacts with short range in the testing dataset is 8274. Therefore, there are $8274 \times 10\% = 827$ qualified predictions for the top 10% limitation. Taking the SVM with ratio 2:1 for example, there are 364 correct predictions (true positive), hence its accuracy (acc) is $364/827 = 44.01\%$. If we observe the top 100% predictions (8274 predictions), the accuracy of the SVM with ratio 2:1 achieves 24.84%. One can see that the SVM with ratio 2:1 has the highest prediction accuracy.

To give a fair assessment for our accuracy 24.84%, we should draw attention to the short-range testing set, which contains 8274 contacts and 147624 noncontacts. That is to say, the probability to find a contact by random exploration is $8274/155898 = 5.31\%$. Therefore, one can see the improvement achieved by our method.

The medium-range and long-range contact predictions have similar results, as shown in Tables 11 and 12. The best results are obtained when the SVM has ratio 2:1. In the medium-range contact prediction, the accuracy of top 100% exploration is 15.68%, which is better than the accru-

Table 10: The experimental results of short-range contact prediction with SVM.

	Ratio 1:1			Ratio 2:1			Ratio 3:1			Ratio 4:1		
Top	N_{corr}	N_{pred}	acc	N_{corr}	N_{pred}	acc	N_{corr}	N_{pred}	acc	N_{corr}	N_{pred}	acc
10%	286	827	34.58%	364	827	44.01%	394	827	47.64%	385	827	46.55%
20%	515	1654	31.14%	619	1654	37.42%	654	1654	39.54%	665	1654	40.21%
30%	718	2482	28.93%	864	2482	34.81%	877	2482	35.33%	884	2482	35.62%
40%	929	3309	28.07%	1075	3309	32.49%	1066	3309	32.22%	1078	3309	32.58%
50%	1131	4137	27.34%	1268	4137	30.65%	1235	4137	29.85%	1256	4137	30.36%
60%	1327	4964	26.73%	1426	4964	28.73%	1398	4964	28.16%	1422	4964	28.65%
70%	1496	5791	25.83%	1591	5791	27.47%	1556	5791	26.87%	1525	5791	26.33%
80%	1646	6619	24.87%	1758	6619	26.56%	1720	6619	25.99%	1525	6619	23.04%
90%	1783	7446	23.95%	1911	7446	25.66%	1871	7446	25.13%	1525	7446	20.48%
100%	1924	8274	23.25%	2055	8274	24.84%	2007	8274	24.26%	1525	8274	18.43%

acy 10529/316441 = 3.33% for randomly exploration (with 10529 contacts and 305912 noncontacts). In the long-range contact prediction, the accuracy of top 100% exploration is 8.23%, which is again better than the accuracy of random exploration (30799/272719 = 1.12%).

5 Conclusion

In this paper, we develop a classification method to predict contacts in a protein, and adopt a new measurement to assess the prediction result. We evaluate three prediction tools (SVM, KNN, and PDA) based on the self-testing for the training set, and then select the best classifier SVM to perform contact prediction on the testing set. According to the experimental results, we provide a more efficient and accurate approach to explore the contacts. Our prediction accuracies for short-range, medium-range, and long-range contacts achieve 24.84%, 15.68%, and 8.23%, which are better than the accuracies 5.31%, 3.33%, and 1.12% achieved by random explorations, respectively. In the future, we would like to improve the prediction accuracy of contacts, and to utilize our results for constructing 3D protein structures, since protein folds are deeply related to these interactions.

References

[1] D. A. adn S. Merler, G. Jurman, R. Visintainer, S. Riccadonna, S. Paoli, and C. Furlanello, "Machine Learning Py - A High-Performance Python/!NumPy Based Package for Machine Learning," 2008. Software available at https://mlpy.fbk.eu/.

[2] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic Acids Research*, Vol. 25, No. 17, p. 33893402, 1997.

[3] P. Bjorkholm, P. Daniluk, A. Kryshtafovych, K. Fidelis, R. Andersson, and T. R. Hvidsten, "Using multi-data hidden Markov models trained on local neighborhoods of protein structure to predict residueresidue contacts," *Bioinformatics*, Vol. 25, No. 10, pp. 1264–1270, 2009.

[4] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," 2001. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[5] J. Cheng and P. Baldi, "Improved residue contact prediction using support vector machines and a large feature set," *BMC Bioinformatics*, Vol. 8(1), pp. 113–121, 2007.

[6] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, Vol. 20, No. 3, pp. 273–297, 1995.

[7] P. Fariselli and R. Casadio, "Prediction of contact maps with neural networks and correlated mutation," *Protein Engineering, Design and Selection*, Vol. 14(11), pp. 835–843, 2001.

[8] G. Faure, A. Bornot, and A. G. de Brevern, "Protein contacts, inter-residue interactions and side-chain modelling," *Biochimie*, Vol. 90(4), pp. 626–639, 2008.

[9] I. Halperin, H. Wolfson, and R. Nussinov, "Correlated mutations: advances and limitations. A study on fusion proteins and on the Cohesin-Dockerin families," *PROTEINS: Structure, Function, and Genetics*, Vol. 63(4), pp. 832–845, 2006.

Table 11: The experimental results of medium-range contact prediction with SVM.

	Ratio 1:1			Ratio 2:1			Ratio 3:1			Ratio 4:1		
Top	N_{corr}	N_{pred}	acc	N_{corr}	N_{pred}	acc	N_{corr}	N_{pred}	acc	N_{corr}	N_{pred}	acc
10%	202	1052	19.20%	361	1052	34.32%	345	1052	32.79%	373	1052	35.46%
20%	354	2105	16.82%	552	2105	26.22%	523	2105	24.85%	562	2105	26.70%
30%	520	3158	16.47%	713	3158	22.58%	679	3158	21.50%	762	3158	24.13%
40%	667	4211	15.84%	864	4211	20.52%	857	4211	20.35%	916	4211	21.75%
50%	805	5264	15.29%	1002	5264	19.04%	1007	5264	19.13%	1070	5264	20.33%
60%	946	6317	14.98%	1153	6317	18.25%	1148	6317	18.17%	1194	6317	18.90%
70%	1071	7370	14.53%	1277	7370	17.33%	1286	7370	17.45%	1315	7370	17.84%
80%	1183	8423	14.04%	1403	8423	16.66%	1405	8423	16.68%	1319	8423	15.66%
90%	1291	9476	13.62%	1530	9476	16.15%	1520	9476	16.04%	1319	9476	13.92%
100%	1402	10529	13.32%	1651	10529	15.68%	1626	10529	15.44%	1319	10529	12.53%

Table 12: The experimental results of long-range contact prediction with SVM.

	Ratio 1:1			Ratio 2:1			Ratio 3:1			Ratio 4:1		
Top	N_{corr}	N_{pred}	acc	N_{corr}	N_{pred}	acc	N_{corr}	N_{pred}	acc	N_{corr}	N_{pred}	acc
10%	248	3079	8.05%	627	3079	20.36%	553	3079	17.96%	558	3079	18.12%
20%	473	6159	7.68%	970	6159	15.75%	913	6159	14.82%	875	6159	14.21%
30%	665	9239	7.20%	1222	9239	13.23%	1186	9239	12.84%	1155	9239	12.50%
40%	824	12319	6.69%	1490	12319	12.10%	1407	12319	11.42%	1387	12319	11.26%
50%	966	15399	6.27%	1701	15399	11.05%	1631	15399	10.59%	1600	15399	10.39%
60%	1107	18479	5.99%	1872	18479	10.13%	1821	18479	9.85%	1795	18479	9.71%
70%	1245	21559	5.77%	2040	21559	9.46%	1993	21559	9.24%	1972	21559	9.15%
80%	1394	24639	5.66%	2216	24639	8.99%	2160	24639	8.77%	2136	24639	8.67%
90%	1529	27719	5.52%	2385	27719	8.60%	2305	27719	8.32%	2286	27719	8.25%
100%	1674	30799	5.44%	2535	30799	8.23%	2461	30799	7.99%	2436	30799	7.91%

[10] N. Hamilton, K. Burrage, M. A. Ragan, and T. Huber, "Protein contact prediction using patterns of correlation," *PROTEINS: Structure, Function, and Genetics*, Vol. 56(4), pp. 679–684, 2004.

[11] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[12] K. M. Misura, D. Chivian, C. A. Rohl, D. E. Kim, and D. Baker, "Physically realistic homology models built with ROSETTA can be more accurate than their templates," *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 103(14), pp. 5361–5366, 2006.

[13] T. Noguchi and Y. Akiyama, "PDB-REPRDB: a database of representative protein chains from the Protein Data Bank (PDB) in 2003," *Nucleic Acids Research*, Vol. 31, No. 1, pp. 492–493, 2003.

[14] O. Olmea and A. Valencia, "Improving contact predictions by the combination of correlated mutations and other sources of sequence information," *Folding and Design*, Vol. 2(3), pp. S25–S32, 1997.

[15] S. Wu and Y. Zhang, "LOMETS: a local meta-threading-server for protein structure prediction," *Nucleic Acids Research*, Vol. 35(10), pp. 3375–3382, 2007.

[16] S. Wu and Y. Zhang, "A comprehensive assessment of sequence-based and template-based methods for protein contact prediction," *Structural Bioinformatics*, Vol. 24, No. 7, pp. 924–931, 2008.