# Protein Remote Homology Detection and Fold Recognition based on Features Extracted from Frequency Profiles

Lei Lin[1, 2], Bin Liu[3], Xiaolong Wang[1, 3], Xuan Wang[3], Buzhou Tang[3]
[1]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
[2]Department of Control Science and Engineering, Harbin Institute of Technology, Harbin, China
[3]Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China
Email: linl@insun.hit.edu.cn, bliu@insun.hit.edu.cn, wangxl@insun.hit.edu.cn, wangxuan@insun.hit.edu.cn, tangbuzhou@gmail.com

***Abstract*—Protein remote homology detection and fold recognition are central problems in bioinformatics. Currently, discriminative methods based on support vector machine (SVM) are the most effective and accurate methods for solving these problems. The performance of SVM depends on the method of protein vectorization, so a suitable representation of the protein sequence is a key step for the SVM-based methods. In this paper, two kinds of profile-level building blocks of proteins, binary profiles and N-nary profiles, have been presented, which contain the evolutionary information of the protein sequence frequency profile. The protein sequence frequency profiles calculated from the multiple sequence alignments outputted by PSI-BLAST are converted into binary profiles or N-nary profiles. The protein sequences are transformed into fixed-dimension feature vectors by the occurrence times of each binary profile or N-nary profile and then the corresponding vectors are inputted to support vector machines. The latent semantic analysis (LSA) model, an efficient feature extraction algorithm, is adopted to further improve the performance of our methods. Experiments with protein remote homology detection and fold recognition show that the methods based on profile-level building blocks give better results compared to related methods.**

***Index Terms*—fold recognition; remote homology detection; Support Vector Machine; Latent semantic analysis, frequency profiles**

## I. INTRODUCTION

Protein homology detection is one of the most intensively researched problems in bioinformatics. Researchers are increasingly depending on computational techniques to classify proteins into functional or structural classes by means of homologies. Most methods can detect homologies at high levels of sequence similarity, while accurately detecting homologies at low levels of sequence similarity (remote homology detection) is still a challenging problem.

Many powerful methods and algorithms have been proposed to detect homology between proteins. Early methods were based on the pairwise similarities between protein sequences. Among those algorithms, the Smith-Waterman dynamic programming algorithm [1] which finds an optimal score for similarity according to a predefined objective function is among the most successful methods. Some heuristic algorithms, such as BLAST [2] and FASTA [3] trade reduced accuracy for improved efficiency. These methods do not perform well for remote homology detection, for the alignment score falls into a twilight zone when the protein sequences similarity is below 35% at the amino acid level [4]. The later methods challenged this problem by incorporating the family information. These methods are based on a proper representation of protein families and can be split into two groups [5]: generative models and discriminative algorithms. Generative models which provide a probabilistic measure of association between a new sequence and a particular family. These methods such as profile hidden Markov Models (HMM) [6], can be trained iteratively in an semi-supervised manner using both positively labeled and unlabeled samples by pulling in close homology and adding them to the positive set [7]. The discriminative algorithms such as Support Vector Machine (SVM) [8] provided state-of-the-art performance with appropriate kernel. In contrast to generative methods, the discriminative algorithms focus on learning a combination of the features that discriminate between the classes. These algorithms are trained in a supervised manner using both positive and negative samples to establish a discriminative model. The first discriminative method is the SVM-Fisher [9], which represents each protein sequence by a vector of Fisher scores. SVM-pairwise [10] is another successful method, in which each protein sequence is represented as a vector of pairwise similarities to all protein sequences in the training set. Many other SVM-based methods also have been proposed such as SVM-k-spectrum [11], Mismatch-SVM [12], SVM-I-sites [13], SVM-n-peptide [14], Monomer-dist [5], GPkernel [15], SVM-LA and SVM-SW [16]. A comparison of SVM-based methods has been performed by Saigo et al. [17].

Sequence homologies are an important source of information about proteins. Multiple sequences alignments of protein sequences contain much information regarding evolutionary processes. This information can be detected by analyzing the output of PSI-BLAST [18, 19]. Since protein sequence frequency profiles are a richer

© 2011 ACADEMY PUBLISHER
doi:10.4304/jcp.6.2.321-328

encoding of protein sequences than the individual sequence, it is of great significance to use such evolutionary information for protein remote homology detection and fold recognition.

In this study, the protein sequence frequency profiles calculated from the multiple sequence alignments outputted by PSI-BLAST are converted into binary profiles or N-nary profiles. The protein sequences are transformed into fixed-dimension feature vectors by the occurrence times of each binary profiles or N-nary profiles and then the corresponding vectors are inputted to support vector machine (SVM). The two methods are further improved by applying an efficient feature extraction algorithm from natural language processing, namely, latent semantic analysis (LSA) [20]. When tested on two SCOP benchmarks, the superfamily and fold recognition problems, the two methods give better results compared to related methods.

## II. METHODS

### A. Data sets

We use a common superfamily benchmark [10] to evaluate the performance of our method for protein remote homology detection. The benchmark contains 54 families and 4352 proteins from SCOP version 1.53 which are extracted from the Astral database [21] and include no pair with a sequence similarity higher than an E-value of 10. Because PSI-BLAST is unable to generate profiles on short sequences, the protein sequences with lengths less than 30 are removed. For each family, the proteins within the family are taken as positive test examples, and the proteins outside the family but within the same superfamily are taken as positive training examples. Negative examples are selected from outside of the superfamily and are separated into training and test sets.

A recently established fold benchmark [15] is used for protein fold recognition. The benchmark contains 3840 proteins from 374 superfamilies and 86 superfamilies are tested. These proteins extracted from SCOP version 1.67 are filtered with Astral database [21] and contain no pair with a sequence similarity more than 95%. The proteins with lengths less than 30 are also removed. For each tested superfamily, there are at least 10 proteins in its positive training and test set. The proteins within one superfamily are taken as positive test samples, while the others in the same fold are taken as positive training examples. The negative test samples are selected from one random superfamily from each of the other folds and the negative training samples are selected from the remaining proteins. Because most of the proteins within a fold have a very low degree of similarity, this fold benchmark is considerably harder than the superfamily benchmark.

### B. Generation of protein sequence frequency profiles

A protein sequence frequency profile can be represented as a matrix $\boldsymbol{M}$, the dimensions of $\boldsymbol{M}$ are $L \times N$, where $L$ is the length of the protein sequence and $N$ is the number of all standard amino acids, which is a constant value of 20. Each element of $\boldsymbol{M}$ is the target frequency which indicates the probability of an amino acid in a specific position of a protein sequence during evolutionary processes. The rows of $\boldsymbol{M}$ are the amino acid frequency profiles. For each row the elements add up to one. Each column of $\boldsymbol{M}$ corresponds to one of the 20 standard amino acids. The protein sequence frequency profiles are calculated from the multiple sequence alignments outputted by PSI-BLAST [18]. The parameter values of PSI-BLAST are set to default except that the number of iterations is set to 10. The database for PSI-BLAST to search against is nrdb90 database (http://www.ebi.ac.uk/~holm/nrdb90) from EBI [22]. A subset of multiple sequence alignments with sequence identity less than 98% is used to calculate the protein sequence frequency profiles. The sequence weight is assigned by the position-based sequence weight method [23]. The calculation of the target frequency is similar to that implemented in PSI-BLAST. Formula (1) is used to calculate the pseudo-count for amino acid $i$ ($g_i$).

$$g_i = \sum_{j=1}^{20} f_i * (q_{ij} / p_j) \tag{1}$$

where $f_i$ is the observed frequency of amino acid $i$, $p_j$ is the background frequency of amino acid $j$, $q_{ij}$ is the score of amino acid $i$ being aligned to amino acid $j$ in BLOSUM62 substitution matrix, which is the default score matrix of PSI-BLAST.

The target frequency is then calculated with the pseudo-count as:

$$Q_i = (\alpha f_i + \beta g_i) / (\alpha + \beta) \tag{2}$$

where $\beta$ is a free parameter set to a constant value of 10 which is initially used by PSI-BLAST and $\alpha$ is the number of different amino acids in a given column minus one.

### C. Converting protein sequence frequency profiles into binary profiles

Two approaches are used to extract the evolutionary information of the protein sequence profiles. The first approach uses binary profiles to approximately represent the protein sequence profiles. The protein sequence profiles are converted into binary profile by a threshold $P_h$. When the frequency of an amino acid is higher than $P_h$, it is converted into an integral value of 1, which means that the specific amino acid can occur in a given position of the protein sequence during evolution. Otherwise it is converted into 0. A substring of amino acid combination is then obtained by collecting the binary profile with non-zero value for each position of the protein sequences. These substrings approximately represent the amino acids that possibly occur at a given sequence position during evolution. Each combination of the twenty amino acids corresponds to a binary profile and vice versa. Binary profiles make up of a profile-based building block of proteins. The process of generating and converting the protein sequence frequency profile into binary profiles is shown in the left part of Fig. 1.

© 2011 ACADEMY PUBLISHER

### D. Converting protein sequence frequency profiles into N-nary profiles

The second approach uses N-nary profiles to extract the evolutionary information of the protein sequence profiles. Because the frequencies of all amino acids are belong to interval [0, 1], this interval can be divided into $N$ equal size intervals. $N$ different integers ranging from 0 to $N$-1 are used to represent the $N$ different equal size intervals respectively (i.e. for $N$=4, interval [0, 1] is divided into 4 equal size intervals: [0, 0.25), [0.25, 0.5), [0.5, 0.75), [0.75, 1] and four integers including 0, 1, 2 and 3 are used to represent the four different equal size intervals respectively). When a given amino acid frequency belongs to a specific interval, the corresponding integer value of the interval is assigned to the amino acid. This process is iterated until each of the 20 standard amino acids is represented as a corresponding integer. Therefore, an amino acid frequency profile is converted into a vector with dimensions of 20, in which each element takes the value from 0 to $N$-1. These elements discriminate the frequencies of the 20 standard amino acids. The bigger the value of the element is, the more probable the corresponding amino acid occurs during evolution. We call such vectors N-nary profiles. The above process is iterated until all amino acid frequency profiles in the protein sequence frequency profile are converted into N-nary profiles. The process of generating and converting the protein sequence frequency profile into N-nary profiles is shown in the right part of Fig. 1.

### E. Chi-square feature selection

Most machine-learning algorithms do not scale well to high-dimensional feature spaces [24]. Thus, it is desirable to reduce the dimension of the feature space by removing non-informative or redundant features. A large number of feature selection methods have been developed for this task, including information gain, mutual information, chi-square and so on. The chi-square algorithm is used in this study because it is one of the most effective feature selection methods in document classification task [25].

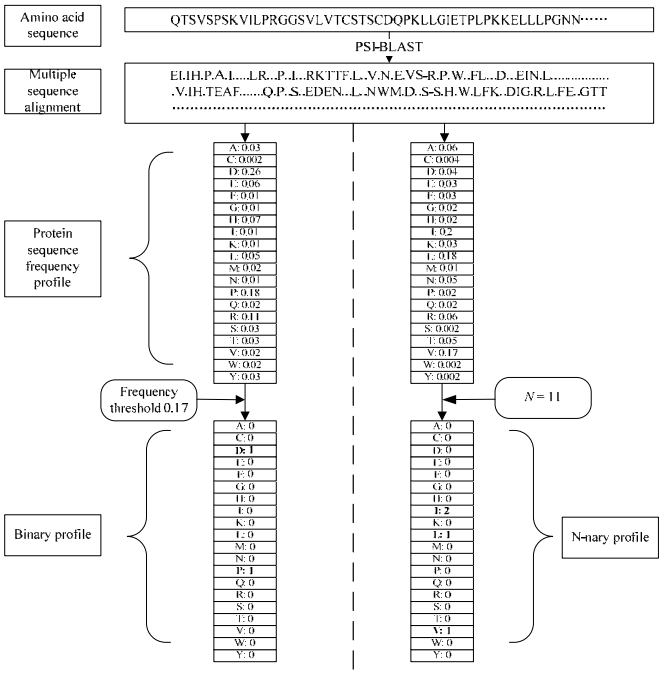



Figure 1. The flowchart of calculating and converting protein sequence frequency profile.

The chi-square algorithm measures the lack of independence between a feature $t$ and a classification category $c$ and can be compared to the chi-square distribution with one degree of freedom to judge extremeness. The chi-square value of feature $t$ relative to category $c$ is defined to be:

$$\chi^2(t,c) = \frac{N \times (A \times D - C \times B)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)} \tag{3}$$

where $A$ is the number of times $t$ and $c$ co-occur, $B$ is the number of times the $t$ occurs without $c$, $C$ is the number of times $c$ occurs without $t$, $D$ is the number of times neither $c$ nor $t$ occurs and $N$ is the total number of protein sequences.

The chi-square statistic has a natural value of zero, if $t$ and $c$ are independent. The category-specific scores of each feature can be combined into one score:

$$\chi^2_{\text{avg}}(\text{t}) = \sum_{i=1}^{m} \text{P}_{\text{r}}(c_i)\chi^2(t,c_i) \tag{4}$$

where $P(c_i)$ is the probability of category $c_i$ and $m$ is the total number of categories.

In this paper, a maximum of 8000 binary profiles or N-nary profiles with highest average features are selected as the "words" of protein sequence language.

### F. Construction of SVM classifiers and classification

In this study, we employ the publicly available Gist SVM package as the implement of SVM. The parameters are used by default of the Gist Package except that the kernel function is set as Radius Basis Function (RBF).

The training proteins are transformed into fixed-dimension feature vectors by the occurrence times of each binary profile or N-nary profiles and then the vectors are inputted to SVM to construct the classifier for a specified class. The test proteins are vectorized in the same way as the training proteins and fed into the classifier constructed for a given class to make separation between the positive and negative samples.

### G. Latent Semantic Analysis

Latent Semantic Analysis (LSA) [20] is combined with our methods to remove noise and compress data. Recently, LSA was introduced in computational biology, it was used to predict the secondary structure of protein [26] and detect protein remote homology [20]. LSA is used to extract and represent the context-usage meaning of words by statistical computations applied to a large corpus of text [27]. The process of LSA is as follows:

Firstly, a word-document matrix $\boldsymbol{W}$ of co-occurrences between words and documents is constructed. The elements of $\boldsymbol{W}$ indicate the numbers of times each word appears in each document, so the dimensions of $\boldsymbol{W}$ are $M \times N$, where $M$ is the total number of words and $N$ is the

© 2011 ACADEMY PUBLISHER

number of given documents. Each word count is normalized to compensate the differences in document lengths and overall counts of different words in the document and collection [27]. Secondly, singular value decomposition is performed on the word-document matrix $\boldsymbol{W}$, as follows:

$$W = USV^T \tag{5}$$

Where $\boldsymbol{U}$ is left singular matrix with dimensions ($M\times K$), $K$ is the total ranks of $\boldsymbol{W}$, $\boldsymbol{S}$ is diagonal matrix of singular values with dimensions ($K\times K$), and $\boldsymbol{V}$ is right singular matrix with dimensions ($N\times K$). Thirdly, the top $R$ ($R$<<Min ($M$, $N$)) dimensions are selected for further processing. The dimensions of reduced matrices $\boldsymbol{U}$, $\boldsymbol{S}$ and $\boldsymbol{V}$ are $M\times R$, $R\times R$ and $N\times R$ respectively.

In this paper, values of $R$ in the range [50, 500] are selected. Five building blocks are treated as the "words", including N-grams [11], patterns [28], motifs [29], binary profiles and N-nary profiles. The protein sequences are viewed as the "documents". Through collecting the weight of each word in the documents, the word-document matrix is constructed and then the latent semantic analysis is performed on the matrix to produce the latent semantic representation vectors of proteins in order to remove noise and compress data. The latent semantic representation vectors are inputted into SVM to give the final results.

TABLE I. COMPARISON AGAINST DIFFERENT METHODS FOR REMOTE HOMOLOGY DETECTION

| Methods | ROC | ROC50 |
|---|---|---|
| PSI-BLAST | 0.6754 | 0.330 |
| SVM-Pairwise | 0.8259 | 0.446 |
| SVM-LA ($\beta$=0.5) | 0.9250 | 0.649 |
| LSTM | 0.9320 | 0.652 |
| SVM-Ngram | 0.7914 | 0.584 |
| SVM-Pattern | 0.8354 | 0.589 |
| SVM-Motif | 0.8136 | 0.616 |
| SVM-Bprofile ($Ph$=0.13) | 0.9032 | 0.681 |
| SVM-N-profile ($N$=11) | 0.9151 | 0.733 |
| SVM-Ngram-LSA | 0.8595 | 0.628 |
| SVM-Pattern-LSA | 0.8789 | 0.626 |
| SVM-Motif-LSA | 0.8592 | 0.628 |
| SVM-Bprofile-LSA ($Ph$=0.13) | 0.9210 | 0.698 |
| SVM-N-profile-LSA ($N$=11) | 0.9402 | 0.736 |

SVM-Ngram, SVM-Pattern, SVM-Motif, SVM-Bprofile and SVM-N-profile refer to the SVM-based methods on the five building blocks: N-grams, patterns, motifs, binary profiles and N-nary profiles respectively. The methods with LSA suffix refer to the corresponding method after latent semantic analysis.

TABLE II. COMPARISON AGAINST DIFFERENT METHODS FOR FOLD RECOGNITION

| Methods | ROC | ROC50 |
|---|---|---|
| PSI-BLAST | 0.5010 | 0.010 |
| SVM-Pairwise | 0.7240 | 0.359 |
| SVM-LA | 0.8340 | 0.504 |
| Mismatch | 0.8140 | 0.467 |
| eMOTIF | 0.6980 | 0.308 |
| GPextended | 0.7530 | 0.371 |
| GPboost | 0.6880 | 0.298 |
| Gpkernel | 0.8440 | 0.514 |
| SVM-Bprofile ($Ph$=0.11) | 0.8042 | 0.644 |
| SVM-N-profile ($N$=9) | 0.7918 | 0.652 |
| SVM-Bprofile-LSA($Ph$=0.11) | 0.8233 | 0.658 |
| SVM-N-profile-LSA ($N$=9) | 0.8226 | 0.658 |

### H. Evaluation methodology

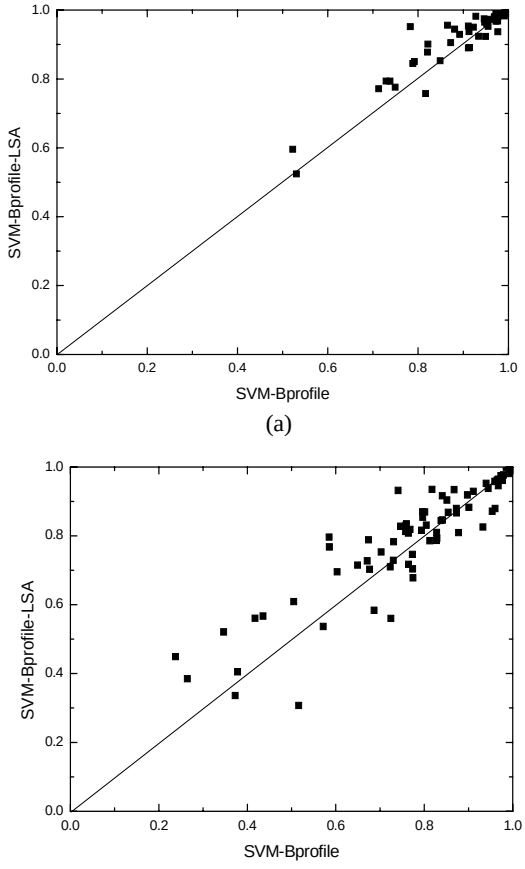
Two methods are used to evaluate the quality of the methods: the Receiver Operating Characteristic (ROC) scores and the ROC50 scores [30]. A ROC score is the normalized area under a curve that plots true positives against false positives for different classification thresholds. A score of 1 denotes perfect separation of positive samples from negative ones, whereas a score of 0 indicates that none of the sequences selected by the algorithm is positive. A ROC50 score is the area under the ROC curve up to the first 50 false positives.

## III. RESULTS AND DISCUSSION

### A. Comparative results of various methods

Table I and table II compare the performance of the methods introduced in this paper against that achieved by a number of previously developed methods for the remote homology detection and fold recognition.

The latent semantic analysis model is adopted to further improve the performance of our methods on the two benchmarks. Fig. 2 and Fig. 3 plot the ROC scores between the methods with LSA and without LSA when binary profiles and N-nary profiles are taken as the building blocks on all test sets for the superfamily benchmark and fold benchmark. When the test sets are in




(a)

(b)

Figure 2. Comparison of the method based on binary profiles with LSA and without LSA. The figure show the SVM-Bprofile's ROC scores plotted against those of the SVM-Bprofile-LSA on the superfamily (a) and fold (b) benchmarks.

© 2011 ACADEMY PUBLISHER

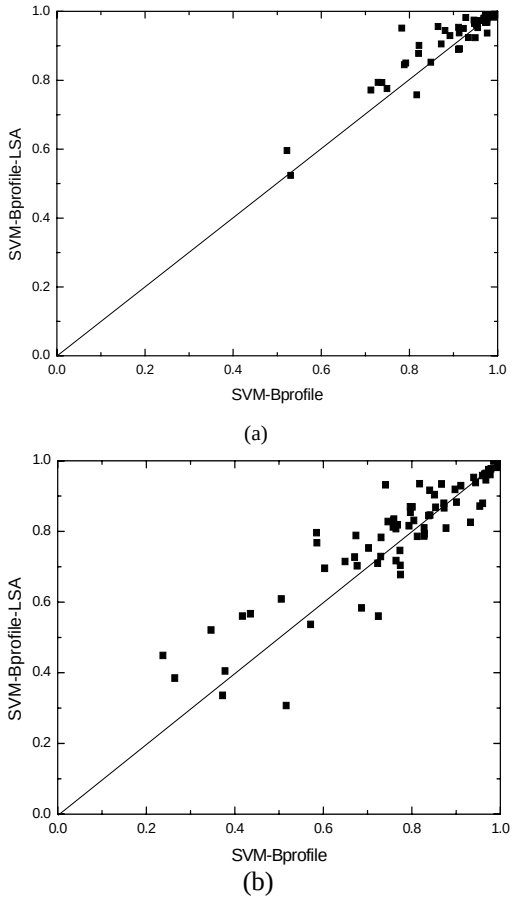
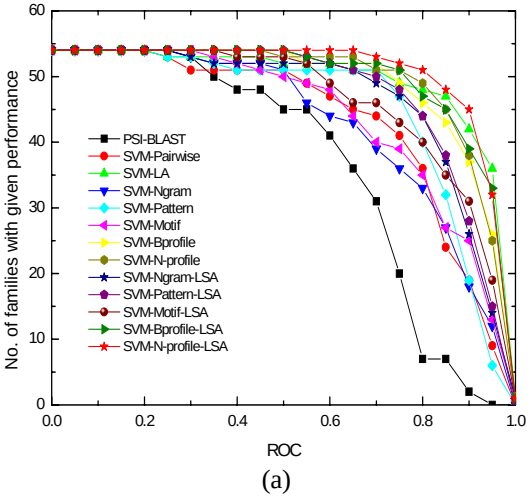



Figure 3. Comparison of the method based on N-nary profiles with LSA and without LSA. The figure show the SVM-N-profile's ROC scores plotted against those of the SVM-N-profile-LSA on the superfamily (a) and fold (b) benchmarks.

the left-upper area, it means that the method labeled by y-axis outperforms the method labeled by x-axis on this test set. Obviously, when binary profiles and N-nary profiles are taken as the basic building blocks, the method with LSA outperforms the method without LSA for both remote homology and fold recognition. As shown in Table I, when N-grams, patterns, and motifs are taken as the building blocks, the methods with LSA also have better performance than the methods without LSA.



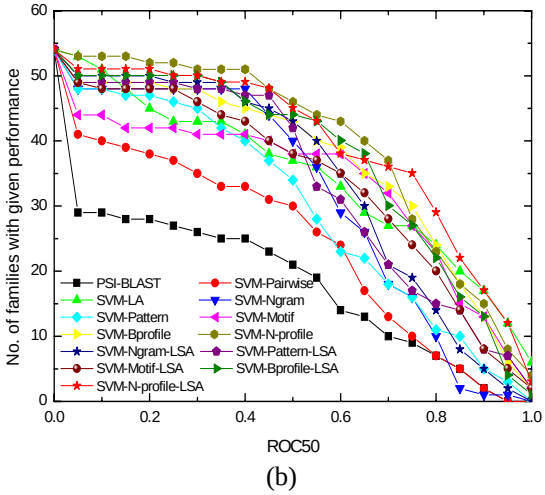
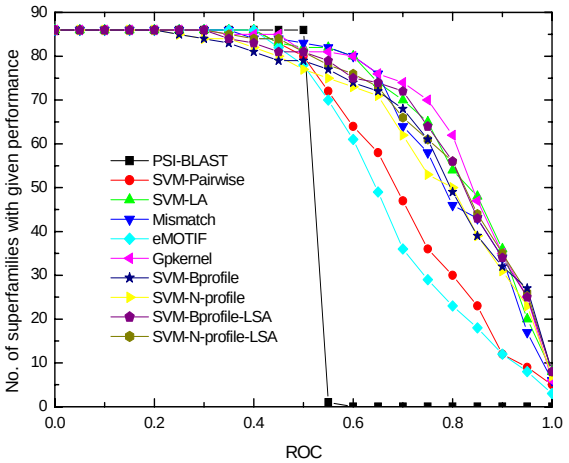




Figure 4. Comparison of some common methods for remote homology detection on superfamily benchmark. The figure plot the total number of families for which a given method exceeds an ROC (a) and ROC50 (b) score threshold.



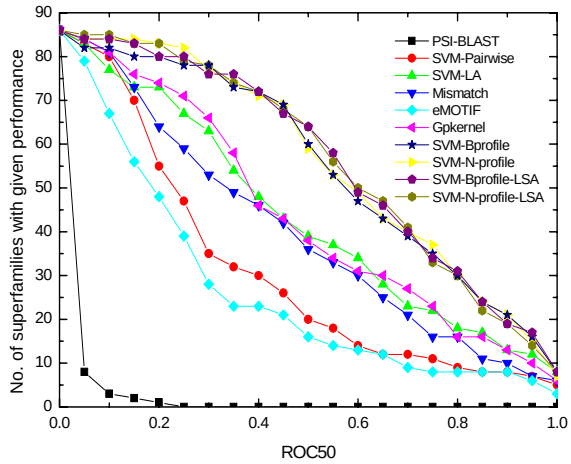




Figure 5. Comparison of some common methods for fold recognition on fold benchmark. The figure plot the total number of superfamilies for which a given method exceeds an ROC (a) and ROC50 (b) score threshold.

© 2011 ACADEMY PUBLISHER

In order to compare our methods with other methods across the different classes, the distributions of ROC and ROC50 scores are plotted in Fig. 4 and Fig. 5. In each graph, a higher curve corresponds to more accurate performance. For remote homology detection, the proposed methods outperform the compared methods in terms of ROC and ROC50. Especially, the ROC50 score of SVM-N-profile-LSA is higher than that of SVM-LA by 8.7 percent. For the fold recognition, our methods are comparable with SVM-LA, Gpkernel and show better performance than the other compared methods. SVM-LA is one of state-of-the-art methods and outperforms many other methods, such as Mismatch-SVM [12] and SVM-Fish [9] as well as FPS [31] and SAM [32]. Therefore, binary profiles and N-nary profiles are two efficient representations of proteins for solving both remote homology detection and fold recognition problems.

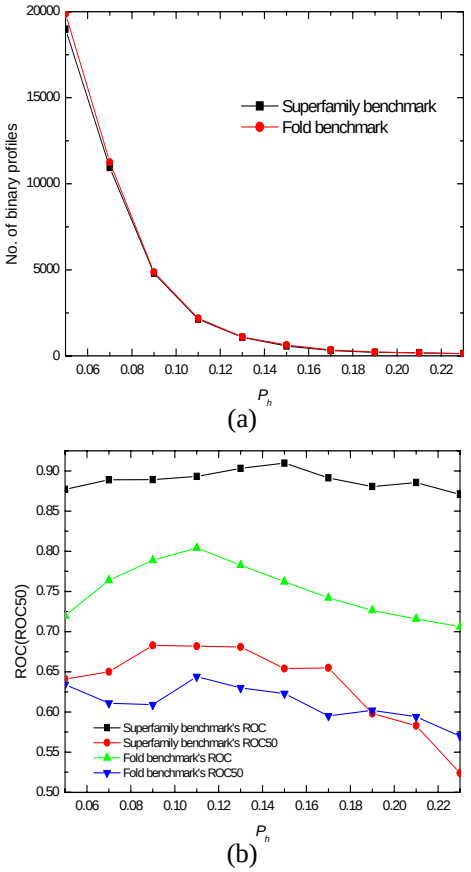



Figure 6. The influence of Ph on the predict performance. Total number of binary profiles (a) and the performance of SVM-Bprofile (b) at different threshold Ph on the superfamily benchmark and the fold benchmark.

### *B. Influences of $P_h$ and N on the performance*

For the method based on binary profiles, the protein frequency profiles are converted into binary profiles with a threshold $P_h$. The total number of binary profiles depends on the size of the database and the value of threshold $P_h$. Since each combination of the twenty amino acids corresponds to one binary profile and vice versa, the total number of binary profiles is $2^{20}$. However, only a small fraction of binary profiles appear in practice as shown in Fig. 6(a). Note that the binary profiles with low occurrence times (<3) are ignored, since these profiles are not statistically significant and may introduce much noise. Since the threshold $P_h$ is a parameter, it needs to be optimized. The results are shown in Fig. 6(b). We surprisingly find that the Ph has not significant influence on the performance. The results show that a small number of binary profiles can contain rich information about evolution.

For the method based on N-nary profiles, the protein sequence frequency profiles are converted into N-nary profiles by using $N$ different integers to represent the corresponding $N$ different equal size intervals. In theory, the total number of N-nary profiles is $N^{20}$. In fact, only a small fraction of binary profiles appear in practice (Fig. 7(a)). The N-nary profiles with low occurrence times (<3) are ignored. The influence of $N$ on the predict performance is shown in Fig. 7(b). We can see that the method based on N-nary profiles performs well for $N$ form 6 to 12. The detection performance increases greatly for $N$ from 2 to 5, since when $N$<6, there are less than 251 N-nary profiles, which contain limited evolutionary information of protein sequence frequency profiles. Note that when $N$=2 the N-nary profiles are equal to the binary profiles with $P_h$ =0.5. The threshold is so high that there are only 21 N-nary profiles containing limited evolutionary information of the protein sequence frequency profiles, which is the reason for the low performance. When $N$>12, the dimension of the feature vector is too high, which is possibly the reason for decreasing in the detection performance.

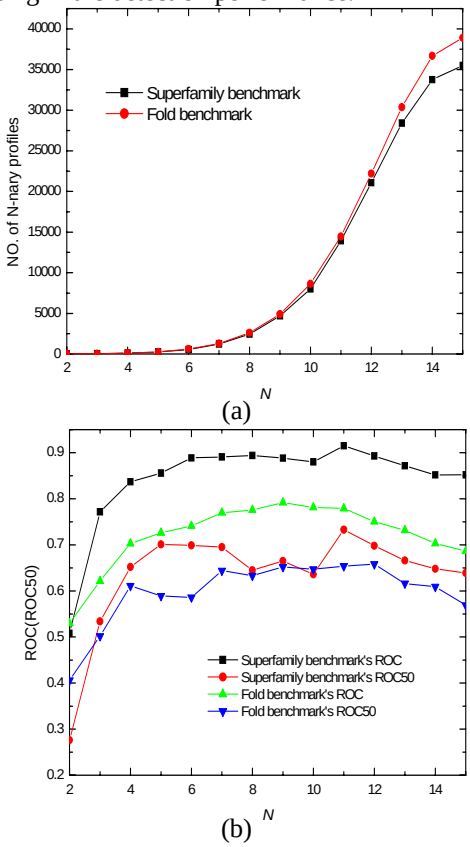



Figure 7. The influence of N on the predict performance. Total number of N-nary profiles (a) and the performance of SVM-N-profile (b) at different threshold Ph on the superfamily benchmark and the fold benchmark.

© 2011 ACADEMY PUBLISHER

## IV. CONCLUSION

In this paper, two simple and efficient building blocks of proteins are introduced, which contain the evolutionary information of protein sequence frequency profiles. The two kinds of building blocks are successfully applied for protein remote homology detection and fold recognition tasks. Experiment results show that the methods based on the two profile-level building blocks outperform all other building-block-based methods. Therefore, binary profile and N-nary profile are suitable profile-level building blocks of the protein sequences. Because the two kinds of building blocks can represent the proteins at sequence-level and residue-level, they can be widely used in many tasks of the computational biology when protein sequence information or amino acid information is needed.

## ACKNOWLEDGMENT

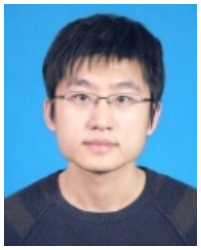
Financial support is provided by the National Natural Science Foundation of China (60973076).

## REFERENCE

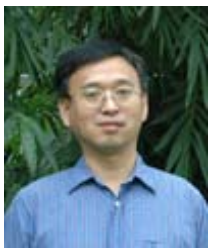
[1] T.F. Smith and M.S. Waterman, "Identification of Common Molecular Subsequences," *J Mol Biol*, vol. 147, no. 1, 1981, pp. 195-197.
[2] S.F. Altschul, et al., "Basic Local Alignment Search Tool," *J Mol Biol*, vol. 215, no. 3, 1990, pp. 403-410.
[3] W.R. Pearson, "Rapid and Sensitive Sequence Comparison with Fastp and Fasta," *Methods Enzymol*, vol. 183, 1990, pp. 63-98.
[4] B. Rost, "Twilight zone of protein sequence alignments," *Protein Eng*, vol. 12, no. 2, 1999, pp. 85-94.
[5] T. Lingner and P. Meinicke, "Remote homology detection based on oligomer distances," *Bioinformatics*, vol. 22, no. 18, 2006, pp. 2224-2231.
[6] K. Karplus, et al., "Hidden Markov Models for Detecting Remote Protein Homologies," *Bioinformatics*, vol. 14, no. 10, 1998, pp. 846-856.
[7] B. Qian and R.A. Goldstein, "Performance of an Iterated T-Hmm for Homology Detection," *Bioinformatics*, vol. 20, no. 14, 2004, pp. 2175-2180.
[8] V.N. Vapnik, *Statistical Learning Theory*, 1998.
[9] T. Jaakkola, et al., "A Discriminative Framework for Detecting Remote Protein Homologies," *J. Comput Biol.*, vol. 7, no. 1-2, 2000, pp. 95-114.
[10] L. Liao and W.S. Noble, "Combining Pairwise Sequence Similarity and Support Vector Machines for Detecting Remote Protein Evolutionary and Structural Relationships," *J. Comput Biol.*, vol. 10, no. 6, 2003, pp. 857-868.
[11] C. Leslie, et al., "The Spectrum Kernel: A String Kernel for svm Protein Classification," *Proc Pacific Symposium on Biocomputing*, 2002, pp. 566-575.
[12] C.S. Leslie, et al., "Mismatch String Kernels for Discriminative Protein Classification," *Bioinformatics*, vol. 20, no. 4, 2004, pp. 467-476.
[13] Y. Hou, et al., "Efficient Remote Homology Detection Using Local Structure," *Bioinformatics*, vol. 19, no. 17, 2003, pp. 2294-2301.
[14] H. Ogul and E.U. Mumcuoglu, "A discriminative method for remote homology detection based on n-peptide compositions with reduced amino acid alphabets," *BioSystems*, vol. 87, no. 1, 2007, pp. 75-81.
[15] T. Håndstad, et al., "Motif kernel generated by genetic programming improves remote homology and fold detection," *BMC Bioinformatics*, vol. 8, 2007, pp. 23.
[16] H. Saigo, et al., "Protein Homology Detection Using String Alignment Kernels," *Bioinformatics*, vol. 20, no. 11, 2004, pp. 1682-1689.
[17] H. Saigo, et al., "Comparison of SVM-Based Methods for Remote Homology Detection," *Genome Informatics*, vol. 13, 2002, pp. 396-397.
[18] S.F. Altschul, et al., "Gapped Blast and Psi-Blast: A New Generation of Protein Database Search Programs," *Nucleic Acids Res*, vol. 25, no. 17, 1997, pp. 3389-3402.
[19] S.E. Dowd, et al., " Windows.Net Network Distributed Basic Local Alignment Search Toolkit (W.Nd-Blast)," *BMC Bioinformatics*, vol. 6, 2005, pp. 93.
[20] Q.W. Dong, et al., "Application of Latent Semantic Analysis to Protein Remote Homology Detection," *Bioinformatics*, vol. 22, no. 3, 2006, pp. 285-290.
[21] S.E. Brenner, et al., "The ASTRAL compendium for sequence and structure analysis," *Nucleic Acids Res*, vol. 28, no. 1, 2000, pp. 254-256.
[22] L. Holm and C. Sander, "Removing near-neighbour redundancy from large protein sequence collections," *Bioinformatics*, vol. 14, no. 5, 1998, pp. 423-429.
[23] S. Henikoff and J.G. Henikoff, "Position-Based Sequence Weights " *J Mol Biol*, vol. 243, no. 4, 1994, pp. 574-578.
[24] A. Andreeva, et al., "Scop Database in 2004: Refinements Integrate Structure and Sequence Family Data," *Nucleic Acids Res*, vol. 32, no. Database, 2004, pp. D226-D229.
[25] Y. Yang and J.A. Pedersen, "A comparative study on feature selection in text categorization," *14th international conference on machine learning*, 1997, pp. 412-420.
[26] M. Ganapathiraju, et al., "Characterization of protein secondary structure, Application of latent semantic analysis using different vocabularies," *IEEE Signal Processing Magazine*, vol. 21, 2004, pp. 78-87.
[27] T.K. Landauer, et al., "Introduction to Latent Semantic Analysis," *Discourse Processes*, vol. 25, 1998, pp. 259-284.
[28] Q. Dong, et al., "A Pattern-Based svm for Protein Remote Homology Detection," *4th international conference on machine learning and cybernetics*, 2005, pp. 3363-3368.
[29] A. Ben-Hur and D. Brutlag, "Remote homology detection: A motif based approach," *Bioinformatics*, vol. 19(Suppl 1), 2003, pp. i26-i33.
[30] M. Gribskov and N.L. Robinson, "Use of Receiver Operating Characteristic (Roc) Analysis to Evaluate Sequence Matching," *Comput Chem*, vol. 20, no. 1, 1996, pp. 25-33.
[31] T.L. Bailey and W.N. Grundy, "Classifying Proteins by Family Using the Product of Correlated P-Values," *3rd international conference on computational molecular biology (RECOMB99)*, 1999, pp. 10-14.
[32] A. Krogh, et al., "Hidden Markov Models in Computational Biology: Applications to Protein Modeling," *J Mol Biol*, vol. 235, no. 5, 1994, pp. 1501-1531.

© 2011 ACADEMY PUBLISHER

**Lei Lin** received his Ph.D. degree in Computer Science and Technology from Harbin Institute of Technology in 2004. He is presently an associate professor of the School of Computer Science and Technology at Harbin Institute of Technology. His research interests include bioinformatics, pattern recognition, and information processing.



**Bin Liu** received his B.E. degree and M.E. degree at Harbin Engineering University and currently is a Ph.D candidate of Harbin Institute of Technology Shenzhen Graduate School. His research interests in computational investigation of sequence–structure–function relationships in proteins and the language model of biological sequence.



**Xiaolong Wang** received his B.E. degree in computer science from the Harbin Institute of Electrical Technology, China, his M.E. degree in computer architecture from Tianjin University, China, and his Ph.D. degree in computer science and engineering from Harbin Institute of Technology in 1982, 1984, and 1989 respectively. He joined Harbin Institute of Technology as an Assistant Lecture in 1984 and became an Associate Professor in 1990. He was a Senior Research Fellow in the Department of Computing, Hong Kong Polytechnic University from 1998 to 2000. Currently, he is a professor of Computer Science at Harbin Institute of Technology. His research interest includes artificial intelligence, machine learning, computational linguistics, and Chinese information processing.

**Xuan Wang** received his Ph.D. degree in Computer Science and Technology from Harbin Institute of Technology in 1997. Currently, he is a professor of Computer Science at Harbin Institute of Technology Shenzhen Graduate School. His research interest includes artificial intelligence, computational linguistics and bioinformatics.

**Buzhou Tang**, Ph.D. candidate in the college of computer science and technology in Harbin Institute of Technology Shenzhen Graduate School. His current research interests include artificial intelligence, bioinformatics and natural language processing.

© 2011 ACADEMY PUBLISHER