Available online at www.sciencedirect.com
ScienceDirect
BioSystems 87 (2007) 75–81
BioSystems
www.elsevier.com/locate/biosystems

A discriminative method for remote homology detection based on *n*-peptide compositions with reduced amino acid alphabets

Hasan Oğul [a,b,*], Erkan Ü. Mumcuoğlu [a]

[a] *Department of Computer Engineering, Başkent University, 06530 Ankara, Turkey*
[b] *Informatics Institute, Middle East Technical University, 06531 Ankara, Turkey*

Received 24 November 2005; received in revised form 20 March 2006; accepted 21 March 2006

Abstract

In this study, *n*-peptide compositions are utilized for protein vectorization over a discriminative remote homology detection framework based on support vector machines (SVMs). The size of amino acid alphabet is gradually reduced for increasing values of *n* to make the method to conform with the memory resources in conventional workstations. A hash structure is implemented for accelerated search of *n*-peptides. The method is tested to see its ability to classify proteins into families on a subset of SCOP family database and compared against many of the existing homology detection methods including the most popular generative methods; SAM-98 and PSI-BLAST and the recent SVM methods; SVM-Fisher, SVM-BLAST and SVM-Pairwise. The results have demonstrated that the new method significantly outperforms SVM-Fisher, SVM-BLAST, SAM-98 and PSI-BLAST, while achieving a comparable accuracy with SVM-Pairwise. In terms of efficiency, it performs much better than SVM-Pairwise. It is shown that the information of *n*-peptide compositions with reduced amino acid alphabets provides an accurate and efficient means of protein vectorization for SVM-based sequence classification.

Keywords: Remote homology; Support vector machine; Protein vectorization; Composition; Reduced alphabet

1. Introduction

Remote homology detection is the problem of detecting homology when there is a weak sequence similarity between structurally homolog proteins. Many studies have been conducted for developing more accurate and faster methods for this task. However, it has been observed that the reliable methods are inefficient in time, whereas the faster methods suffer from the low prediction accuracy. Owing to the steady increase in the amount of protein data, developing methods that achieve the required level of accuracy with a reasonable cost of time has become an urgent need.

The early methods for homology detection were based on the pairwise sequence similarity inferred by dynamic-programming-based sequence alignment (Smith and Waterman, 1981). While the dynamic-programming method finds an optimal score for similarity according to a predefined objective function, it suffers from long computation times for relatively long sequences. To speed up the alignment, some heuristic methods, such as BLAST (Altschul et al., 1990), have been developed to find a near-optimal alignment within a reasonable time. Although these methods are very successful in the search of homolog proteins, they do not perform well for the detection of remote homologies since the alignment score falls into a twilight zone when

* Corresponding author. Tel.: +90 3122341010; fax: +90 3122341051.
E-mail address: hogul@baskent.edu.tr (H. Oğul).

0303-2647/$ – see front matter © 2006 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.biosystems.2006.03.006

the sequence identity is below 35% (Rost, 1999). The later methods have incorporated the family information to detect the more distant homologies and achieved approximately three times as accurate results as simple pairwise comparison methods (Park et al., 1998). These methods are based on the similarity statistics derived upon more than one homolog example, that is, all statistical information is generated from a set of sequences that are known or posited to be evolutionary related to another. These probabilistic methods are often called as generative because they induce a probability distribution over the protein family and try to generate the unknown protein as a new member of the family from this stochastic model. Further improvements have also been achieved by iteratively collecting homolog proteins from a large database and incorporating the resulting statistics into a central model (Altschul et al., 1997; Karplus et al., 1998). The main problem with generative approaches is the fact that they produce excessive false positives, that is, they report a number of homologs though they are not.

The recent works on remote homology detection have begun to use a discriminative framework to make separation between homolog (positive) and non-homolog (negative) classes. In contrast to generative methods, the discriminative methods focus on learning a combination of the features that discriminate between the classes. These methods try to establish a model that differentiates between positive and negative examples. In other words, non-homologs are also taken into account. The first discriminative approach (SVM-Fisher) represented each protein by a vector of Fisher scores extracted from a Hidden Markov model profile constructed for a protein family and utilized support vector machines (SVM) to classify the protein with those feature vectors (Jaakola et al., 2000). A recent and more successful work, called SVM-Pairwise (Liao and Noble, 2003), combined the sequence similarity with the SVMs to discriminate between positive and negative examples. In SVM-Pairwise, both the training and test sets include positive and negative examples. This method has been tested for dynamic-programming-based alignment scores and BLAST scores. Note that the latter one is referred as SVM-BLAST in the following sections. SVM-Pairwise approach is among the best methods in terms of accuracy, but it suffers from computational inefficiency since the alignment takes too much time for long sequences. In general, discriminative methods are more successful than generative methods in terms of separation accuracy between true positives and false positives. However, the training phase requires long times with conventional workstations, which makes them inappropriate to use in practice. Thus, more efficient methods are required while preserving the classification accuracy.

In the present study, we use n-peptide compositions of a protein to encode it over a discriminative framework based on SVM. Amino acid composition, as a special case of n-peptide composition for $n = 1$, has been widely used on many problems regarding protein classification (Bahar et al., 1997; Zhang et al., 1995). Dipeptide composition ($n = 2$) has also been shown to be valuable information in the determination of protein classes (Bhasin et al., 2005; Yu et al., 2004). However, composition of longer n-peptides could not be used efficiently since the time and memory requirements exponentially increase with the value of n. In this study, we present a way of restricting memory requirement by reducing the amino acid alphabet for larger values of n. We also use a hash structure for accelerating the search of n-peptides. The system allows the inclusion of n-peptides up to $n = 6$ with increasing accuracy. We have integrated the new compositional encoding method into SVM to perform protein family classification tests on a subset of SCOP family database (Murzin et al., 1995) and compared our results against those given by recent methods; SAM-98 (Karplus et al., 1998), PSI-BLAST (Altschul et al., 1997), SVM-Fisher (Jaakola et al., 2000), SVM-BLAST and SVM-Pairwise (Liao and Noble, 2003). The new method achieves a significantly better accuracy than all given methods except SVM-Pairwise. The new method's accuracy is comparable to that of SVM-Pairwise. In terms of computational efficiency, the new method performs much better than SVM-Pairwise.

2. Materials and methods

2.1. Data set

We used the data set provided by Liao and Noble, available at http://www1.cs.columbia.edu/compbio/svm-pairwise, in our experiments. The methods were tested to see their ability to classify proteins into families on the given subset of SCOP1.53 family database (Murzin et al., 1995). The data set contains 4352 protein domains including no pair with a sequence similarity higher than an E-value of 10^{-25}. The training and test sets are separated as done in the previous works, resulting with 54 families to test. 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 considered positive training examples. Negative examples are selected from outside of the superfamily and are randomly separated into training and test sets in the same ratio as the positive examples.

2.2. Discriminative model

In discriminative homology detection methods, there are two main phases: training and testing. The training phase constructs a machine learning classifier for the specified family, and the testing phase uses this classifier to decide whether the test protein is belonging to this family or not. Both phases require the extraction of some informative features from the protein sequence and the representation of these features by a fixed-length feature vector. Fig. 1 gives an overview of the discriminative homology detection approach.

2.3. Protein vectorization

Each protein must be represented by a fixed-length feature vector to be fed into a machine learning classifier. We represent proteins by their n-peptide compositions. For each value of n, corresponding feature vector contains the fraction of each possible n-length substring in the sequence. For example, the feature vector refers to amino acid composition for $n = 1$, and dipeptide composition for $n = 2$. The number of dimensions in the feature vector corresponding to n-peptide composition is 20^n. The memory space complexity of the training step then becomes $O(k20^n)$, where k is the number of proteins in the training set. This leads to the formation of high-dimensional feature vectors even for small values of n, which makes the system difficult to implement with conventional memory resources. To overcome this problem, we gradually reduce the size of the amino acid alphabet for increasing values of n such that the resulting vector for each n-peptide composition will have a dimension lower than a constant value of t, hence getting an upper bound on the space complexity by $O(kt)$. In other words, we use an alphabet size of r that satisfies the condition $r^n < t$ for n-peptide composition. Not only providing an efficient space complexity, this scheme also allows the evaluation of possible mismatches in longer n-peptides, which is a natural case in the evolution of proteins. Table 1 gives the reduced amino acid alphabet sizes and resulting feature vector dimensions for different values of t to be used in the construction of n-peptide compositions.

Another problem with the use of n-peptide compositions is the exponential time complexity of the protein vectorization step. A naive algorithm that searches all possible n-peptides in a protein sequence of length m has a time complexity of $O(m20^n)$. We use a hash structure indexed by a sorted array of all possible n-peptides and sequentially traced the sequence to

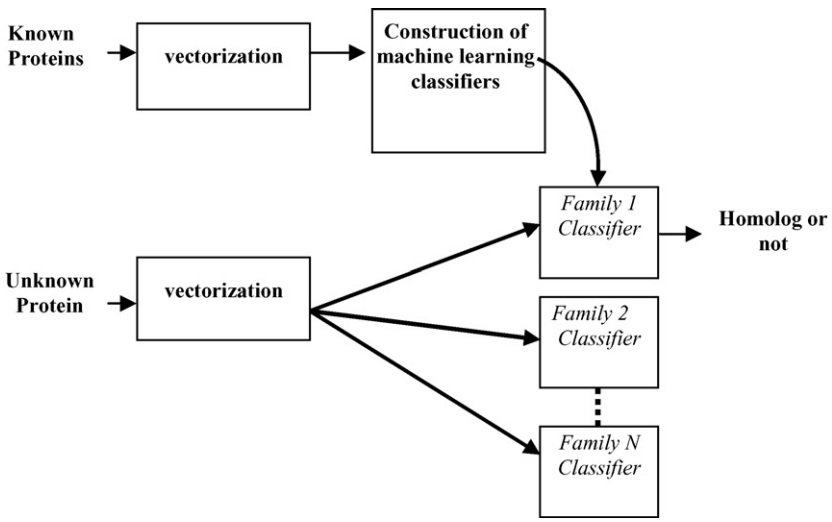

Fig. 1. Discriminative homology detection model.

Table 1
The amino acid alphabet sizes and resulting feature vector dimensions used for each n-peptide composition with varying thresholds of dimension

n	$t = 1000$		$t = 5000$		$t = 10000$	
	Alphabet size	Vector dimension	Alphabet size	Vector dimension	Alphabet size	Vector dimension
1	20	20	20	20	20	20
2	20	400	20	400	20	400
3	10	1000	15	3375	20	8000
4	5	625	8	4096	10	10000
5	3	243	5	3125	6	7776
6	3	729	4	4096	4	4096

Table 2
Reduced amino acid alphabets used in our experiments

Size	Alphabet
20	L V I M C A G S T P F Y W E D N Q K R H
15	(LVIM) C A G S T P (FY) W E D N Q (KR) H
8	(LVIMC) (AG) (ST) P (FYW) (EDNQ) (KR) H
6	(LVIM) (AGST) (PHC) (FYW) (EDNQ) (KR)
5	(LVIMC) (AGSTP) (FYW) (EDNQ) (KRH)
4	(LVIMC) (AGSTP) (FYW) (EDNQKRH)
3	(LVIMCAGSTP) (FYW) (EDNQKRH)
2	(LVIMCAGSTPFYW) (EDNQKRH)

update the counts of the observed n-peptide. With this scheme, the time complexity is reduced to $O(mn)$. Since we use only small values of n, the time complexity can be simply regarded as $O(m)$. The system has been implemented by standard library functions of Perl language.

2.4. Reduced amino acid alphabets

We use the reduced amino acid alphabets provided by Murphy et al. (2000) in our method. These alphabets have been produced using statistical techniques based on the information of certain BLOSUM matrices and justified by well-known biochemical amino acid classes. Table 2 lists the reduced amino acid alphabets used in our experiments.

2.5. Construction of machine learning classifiers

As illustrated in Fig. 1, the discriminative homology detection system requires the construction of a machine learning classifier for each family to make separation between the homolog and non-homolog examples. We employ SVMs for this purpose. SVMs are powerful binary classifiers that work based on the structural risk minimization principle. An SVM non-linearly maps its multi-dimensional input space into a higher dimensional feature space. In this feature space a linear classifier is constructed. To train the SVMs, the Gist software, available at http://www.cs.columbia.edu/compbio/svm, is used. Since the main novelty of this study is introducing a new vectorization scheme, the SVM parameters are selected in the same way with the previous methods in order to make a fair comparison. We use a radial basis kernel function;

$$K'(X, Y) = \mathrm{e}^{-(K(X,X)-2K(X,Y)+K(Y,Y)/2\sigma^2)} + 1$$

where $K(.,.)$ is the normalized base kernel that acts as a similarity score between the pair of input vectors X and Y and the width σ is the median Euclidean distance from any positive training example to the nearest negative example. Since the separating hyperplane of SVM is required to pass from the origin, the constant 1 is added to the kernel so that the data goes away from the origin. In addition, an asymmetric soft margin is implemented by adding to the diagonal of the kernel matrix a value $0.02^{*}\rho$, where ρ is the fraction of training set proteins that have the same label with the current protein, as done in the previous SVM classification methods (SVM-Pairwise, SVM-BLAST, SVM-Fisher).

2.6. Classification

The test samples are vectorized in the same way with the training samples and fed into the SVM constructed for the specified family. The SVM output is a list of discriminant scores corresponding to each protein in the test set. The discriminant score indicates a predicted level of homology with that method. The system classifies the proteins with a discriminant score higher than a threshold value (generally zero) as homologs and the others as non-homologs.

2.7. Evaluation of methods

As all classification tasks do, the homology detection methods have to deal with the trade-off between specificity (the ability to reject false positives) and sensitivity (the ability to detect true positives). For the cases in which the positive and negative examples are not evenly distributed, the best way to evaluate the trade-off between the specificity and sensitivity is to use a receiver operating characteristics (ROC) curve (Gribskov and Robinson, 1996). A ROC score may be defined as the area under the ROC curve, where the ROC curve is plotted as the number of true positives as a function of false positives for varying classification thresholds. A score of 1 indicates that the positives are perfectly separated from negatives whereas the score of 0 yields that no positives are reported.

3. Results

3.1. Accuracy

As a result of the SVM classification tests, the ROC scores were calculated for each family included in the experimental setup. The average values of ROC scores resulted of six different methods, including the present method (SVM-n-peptide), are listed in Table 3.

As shown in the table, the system's performance increases until the inclusion of n-peptide composition with $n=6$ and degrades after n becomes larger than 8. With a threshold of 5000 in vector dimensions, the number of letters in the simplified alphabet for $n>7$ must be reduced to 2. Since an alphabet size of 2 does not carry any information about the protein evolution, this is possibly the reason for decrease in the accuracy after $n>7$. A comparison between the cases for different dimension thresholds applied is also presented in the table. According to the results, when the threshold is increased to 10,000 no improvement is observed. On the other hand, the accuracy reduces when the threshold is lowered into 1000. Therefore, the threshold value of 5000 seems to be a good selection for satisfying both the accu-

Table 3
Average ROC scores obtained from 54 SCOP families

Method	Average ROC score
PSI-BLAST	0.582
SAM-98	0.651
SVM-Fisher	0.676
SVM-BLAST	0.816
SVM-Pairwise	0.892
SVM-*n*-peptide	
$n=1$	0.814
$n=1$-2	0.851
$n=1$-3	0.879
$n=1$-4, $t=5000$	0.879
$n=1$–6, $t=1000$	0.869
$n=1$–6, $t=5000$	0.890
$n=1$–6, $t=10000$	0.890
$n=1$–7, $t=5000$	0.890
$n=1$–8, $t=5000$	0.881
$n=1$–10, $t=5000$	0.879

racy and efficiency requirement of the system. The table also demonstrates the surprising success of amino acid composition ($n=1$) alone in homology detection task. The accuracy achieved with the use of amino acid composition is better than many of the more complicated methods included in our comparative study.

For further investigation of the results, the methods are compared by their relative performances using the plots of the number of families for which a given method exceeds a threshold ROC score. Fig. 2 compares the performance of new method for $n=1$–6 and $t=5000$ with the existing methods. Both average ROC scores and the comparison plot demonstrate that the new method significantly outperforms all given methods except SVM-Pairwise, while being comparable with it. To explore the statistical significance of differences between the results, paired *T*-tests were carried out between SVM-based methods with a *p*-value threshold of 0.05. According to Table 4, a significant difference SVM-*n*-peptide and SVM-BLAST is observed. The difference between SVM-*n*-peptide and SVM-Fisher is more apparent. On the other hand, there is no significant difference between SVM-*n*-peptide and SVM-Pairwise.

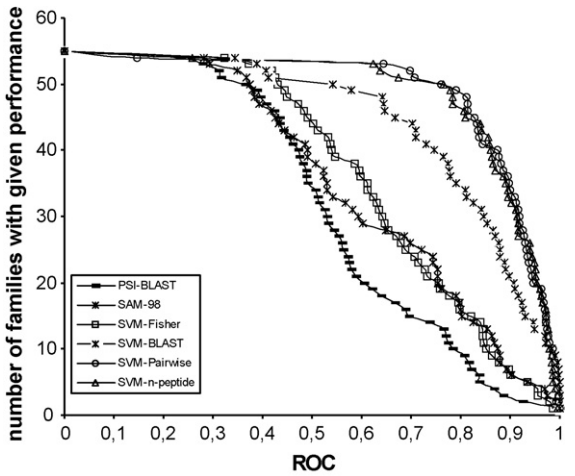

Fig. 2. Relative performances of different classification methods are depicted by the plots of the number of families for which a given method exceeds a threshold ROC score.

Table 4
Results of paired *T*-tests based on the ROC scores achieved by SVM-based remote homology detection methods

p-values	SVM-*n*-peptide	SVM-Pairwise	SVM-BLAST	SVM-Fisher
SVM-*n*-peptide		0.37	5.0E-3	3.9E-12
SVM-Pairwise			2.7E-3	3.3E-13
SVM-BLAST				3.8E-10

A family-by-family comparison between the results of SVM-Pairwise and SVM-*n*-peptide is provided in Table 5. The table shows that two methods are not only competitive but also complementary to the each other. When the results are investigated in superfamily level, it is observed that SVM-*n*-peptide is more successful for all families in homeodomain-like proteins (1.4.1.×), nucleic acid-binding proteins (2.38.4.×), viral coat and capsid proteins (2.9.1.×), glycosyltransferases (3.1.8.×) and P-loop containing nucleotide triphosphate hydrolases (3.32.1.×). This result would be useful when selecting the appropriate method in any application that requires an automated or semi-automated search in SCOP database.

In order to see the effect of different amino acid groupings, we also applied the schemes provided by Li et al. (2003) and Liu et al. (2002) for reducing alphabets. For the homology detection tests with $n=1$–6 and $t=5000$, former scheme provided an average ROC score of 0.893 with 0.132 standard deviation and the latter one provided 0.889 average ROC score with 0.134 standard deviation. Although we do not observe a statistically significant difference between them ($p>0.05$ for all paired *T*-tests), the ROC score deviations with the alphabets of Murphy et al. (2000) is lower than those with other alphabets.

3.2. Computational efficiency

Computational efficiency is another important aspect in the evaluation of methods. The crucial step in the SVM system we used is the vectorization of proteins. The vectorization step has a complexity of $O(mp)$ in

Table 5
Family-by-family comparison of SVM-Pairwise and SVM-*n*-peptide based on the ROC scores obtained from each family test

Family	ROC score	
	SVM-Pairwise	SVM-*n*-peptide
1.27.1.1	0.971	0.948
1.27.1.2	0.918	0.962
1.36.1.2	0.935	0.916
1.36.1.5	0.976	0.961
1.4.1.1	0.968	0.977
1.4.1.2	0.814	0.976
1.4.1.3	0.944	0.973
1.41.1.2	0.999	0.994
1.41.1.5	0.998	0.990
1.45.1.2	0.971	0.810
2.1.1.1	0.978	0.892
2.1.1.2	0.994	0.985
2.1.1.3	0.985	0.976
2.1.1.4	0.974	0.894
2.1.1.5	0.832	0.807
2.28.1.1	0.815	0.637
2.28.1.3	0.829	0.865
2.38.4.1	0.697	0.766
2.38.4.3	0.707	0.779
2.38.4.5	0.877	0.916
2.44.1.2	0.146	0.259
2.5.1.1	0.925	0.896
2.5.1.3	0.896	0.783
2.52.1.2	0.643	0.783
2.56.1.2	0.844	0.855
2.9.1.2	0.874	0.951
2.9.1.3	0.970	0.996
2.9.1.4	0.918	0.984
3.1.8.1	0.963	0.987
3.1.8.3	0.931	0.973
3.2.1.2	0.838	0.887
3.2.1.3	0.898	0.859
3.2.1.4	0.964	0.939
3.2.1.5	0.932	0.914
3.2.1.6	0.912	0.903
3.2.1.7	0.909	0.955
3.3.1.2	0.937	0.916
3.3.1.5	0.917	0.943
3.32.1.1	0.946	0.952
3.32.1.11	0.880	0.973
3.32.1.13	0.836	0.938
3.32.1.8	0.901	0.912
3.42.1.1	0.886	0.840
3.42.1.5	0.811	0.624
3.42.1.8	0.760	0.674
7.3.10.1	0.986	0.985
7.3.5.2	0.996	0.987
7.3.6.1	0.998	0.978
7.3.6.2	0.994	0.965
7.3.6.4	0.992	0.995
7.39.1.2	0.928	0.863
7.39.1.3	0.990	0.870
7.41.5.1	0.791	0.841
7.41.5.2	0.943	0.860
Average	0.892	0.890
S.D.	0.133	0.125

SVM-Fisher, where m is the length of the longest training set sequence and p is the number of parameters used in the profile HMM. SVM-Pairwise and SVM-BLAST calculate all pairwise similarity scores between the target sequence and the sequences in the training set. Each similarity calculation is $O(m^2)$ in SVM-Pairwise and $O(m)$ in SVM-BLAST. Thus, total vectorization time is $O(km^2)$ for SVM-Pairwise and $O(km)$ for SVM-BLAST, where k is the number of proteins in the training set. Our vectorization scheme has a time complexity of $O(m)$ as described in the Methods section. The complexity analysis reveals that our method is more efficient than all other SVM methods described in this paper.

An empirical comparison in terms of computation time may be invalid since much of the work in our implementation contains file processing owing to large amount of data that cannot be handled by memory. However, to make an intuition, we can report that all training time is at most 1 h for a family with SVM-*n*-peptide, whereas it takes at least 20 days with SVM-Pairwise in a workstation having 2 GHz CPU and 2 GB memory. In Table 6, we provided some empirical test results that demonstrate the computation times for the predictions of selected proteins with varying lengths. The tests were performed on two families having the lowest and largest number of training samples in the dataset. As shown, the computation time drastically increases with the protein length when the SVM-Pairwise method is used. On the other hand, we observe only a slight increase in the prediction time of SVM-*n*-peptide for longer pro-

Table 6
Computation times for the predictions of selected proteins with SVM-Pairwise and SVM-*n*-peptide

Test protein (PDB ID)	Sequence length	Test family (SCOP ID)	Number of training samples	SVM-Pairwise (in s)	SVM-*n*-peptide (in s)
1a7f_1	21	7.3.10.1	434	61	18
		2.1.1.3	4008	290	58
1ulo	152	7.3.10.1	434	195	20
		2.1.1.3	4008	1632	63
1914	208	7.3.10.1	434	251	20
		2.1.1.3	4008	2189	65
1qcxa	359	7.3.10.1	434	405	22
		2.1.1.3	4008	3730	67
1aco_2	527	7.3.10.1	434	577	23
		2.1.1.3	4008	5439	70
1yge_1	690	7.3.10.1	434	744	24
		2.1.1.3	4008	7112	71
1eula	994	7.3.10.1	434	1054	26
		2.1.1.3	4008	10217	75

PDB: Protein Data Bank (Berman et al., 2000), SCOP: Structural Classification Of Proteins (Murzin et al., 1995).

teins. The computation time scales almost linearly with the number of training samples in both methods. However, the prediction time of SVM-*n*-peptide still remains around 1 min for the worst case in the table, whereas SVM-Pairwise spends nearly 7 h for the same prediction. The results apparently show that the new system substantially reduces the computation time needed for both training and testing phases while preserving the overall accuracy achieved by SVM-Pairwise. Furthermore, SVM-*n*-peptide becomes much more practical to use for larger datasets and longer sequences.

4. Discussion

Selecting a proper feature representation scheme is an important step in classification systems. The problem arises more seriously in protein classification tasks owing to the fact that protein sequences are of varying lengths. In this study, the information of *n*-peptide compositions has been successfully applied for remote protein homology detection task over a discriminative framework employing SVMs. The representation with *n*-peptide compositions is quite simple; it does not require sequence alignments, profile construction or motif search. The study has shown that the use of reduced amino acid alphabets for longer *n*-peptides provides an efficient and accurate way of protein vectorization for sequence-based protein classification systems. The use of reduced amino acid alphabets not only provides an efficient representation for amino acid composition and local sequence order effects of proteins but also gives the opportunity to evaluate the possible mismatches between the sequences.

The new method, which we call as SVM-*n*-peptide, has been tested for SCOP family classification on a common benchmarking data set. Comparison with the existing methods reveals that the present method significantly outperforms the others except SVM-Pairwise, while being comparable with it. On the other hand, a remarkable improvement has been achieved in terms of computational efficiency in comparison with SVM-Pairwise.

We believe that the method that we have introduced here can be applicable to other protein classification tasks as well.

References

Altschul, S., Gish, W., Miller, W., Myers, E.W., Lipman, D., 1990. A basic local alignment search tool. J. Mol. Biol. 251, 403–410.

Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W., Lipman, D., 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 25, 3389–3402.

Bahar, I., Atilgan, A.R., Jernigan, R.J., Erman, B., 1997. Understanding the recognition of protein structural classes by amino acid composition. Proteins 29, 172–185.

Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E., 2000. The protein data bank. Nucleic Acids Res. 28, 235–242.

Bhasin, M., Garg, A., Raghava, G.P.S., 2005. PSLpred: prediction of subcellular localization of bacterial proteins. Bioinformatics 21, 2522–2524.

Gribskov, M., Robinson, N.L., 1996. Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. Comp. Chem. 20, 25–33.

Jaakola, T., Diekhans, M., Haussler, D., 2000. A discriminative framework for detecting remote protein homologies. J. Comput. Biol. 7, 95–114.

Karplus, K., Barrett, C., Hughey, R., 1998. Hidden Markov models for detecting remote protein homologies. Bioinformatics 14, 846–856.

Liao, L., Noble, W.S., 2003. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. J. Comput. Biol. 10, 857–868.

Li, T., Fan, K., Wang, J., Wang, W., 2003. Reduction of protein sequence complexity by residue grouping. Protein Eng. 16, 323–330.

Liu, X., Liu, D., Qi, J., Zheng, W., 2002. Simplified amino acid alphabets based on deviation of conditional probability from random background. Phy. Rev. E 66, 021960.

Murphy, L.R., Wallqvist, A., Levy, R.M., 2000. Simplified amino acid alphabets for protein fold recognition and implications for folding. Protein Eng. 13, 149–152.

Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C., 1995. SCOP: a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Biol. 247, 536–540.

Park, J., Karplus, K., Barrett, C., Hughey, R., Haussler, D., Hubbard, T., Chothia, C., 1998. Sequence comparisons using multiple sequences detect tree times as many remote homologues as pairwise methods. J. Mol. Biol. 284, 1201–1210.

Rost, B., 1999. Twilight zone of protein sequence alignments. Protein Eng. 12, 85–94.

Smith, T.F., Waterman, M.S., 1981. Identification of common molecular subsequences. J. Mol. Biol. 147, 195–197.

Yu, C., Lin, C., Hwang, J., 2004. Predicting subcellular localization of proteins for Gram-negative bacteria by support vector machines based on *n*-peptide compositions. Protein Sci. 13, 1402–1406.

Zhang, C.T., Chou, K.C., Maggiora, G.M., 1995. Predicting protein structural classes from amino acid composition: application of fuzzy clustering. Protein Eng. 8, 425–435.