Journal of Biological Sciences 5 (): cccc, 2005
ISSN 1727-3048

Representing Protein Sequence with Low Number of Dimensions

Nazar Zaki[1] and Safaai Deris[2]
[1]College of IT, UAE University, Al Ain 17555, UAE
[2]SPS, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia

Abstract: This research work introduces a simple method based on representing protein sequence by fix dimensions of the length three. We present hidden Markov model combining scores method. Three scoring algorithms are combined to represent protein sequence of amino acids for better remote homology detection. We tested the method on the SCOP version 1.37 dataset. The results show that, with such a simple representation, we are able to achieve superior performance to previously presented protein homology detection methods while achieving better computational efficiency.

Keywords: Support vector machine, hidden Markov model, protein homology detection

INTRODUCTION

Protein remote homology detection is the task of classifying protein sequence into one predefined family. Effective representation of the protein sequence is a key issue in detecting remote protein homology. Much research has been done on protein homology detection and classification. Dynamic programming based alignment tools such as Smith and Waterman [1] and their approximation such as FASTA [2] and BLAST [3] have been widely used by biologist around the world. Statistical model based methods have also been developed such as Profile [4] and hidden Markov models (HMM) [5, 6]. Iterative methods such as PSI-BLAST [7] and SAM [8] improved upon profile-based methods. The SVM-Fisher method [9] which combines an iterative HMM training scheme with a discriminative algorithm known as Support Vector Machine (SVM) [10, 11] is currently among the most accurate known methods for detecting remote protein homology. SVM-Fisher begins by training a generative HMM for a protein family and then, using the model to extract a representation for each protein sequence in the form of *sufficient statistics*. The sufficient statistics are then treated to produce an analogous quantity known as the *Fisher score*. SVMs in conjunction with the Fisher scores are then used to discriminate between protein families.

Recently, Leslie *et al*. [12] introduced a class of string kernels, called *mismatch kernels*, to be used with support vector machines. In this method, a kernel function measures the similarity between a pair of inputs, and defines an inner product in an implicit feature space for the SVM optimization problem. The features used by the mismatch kernel are the set of all possible subsequences of amino acids of a fixed length. The mismatch method has achieved state-of-the-art performance for protein classification. However, such representation suffers high dimensionality problem which cause longer running time and space to compute the kernel entries. In this work, we introduce a simple but yet effective method based on representing the protein sequence by only three dimensions using the concepts introduced by the SVM-Fisher method. We combine techniques from machine learning and information extraction to capture the evolutionary relationships between protein sequences. We directed our work to achieve better protein homology detection by representing protein sequence with minimum fixed-length vectors possible. This is done through incorporating the maximum biological information and evolutionary relationships into a minimum number of dimensions. The kernel matrix is then calculated efficiently and used as a source of information for the SVM to achieve good classification performance.

MATERIALS AND METHODS

Fig. 1, shows the overview of the proposed method which we call as SVM-HMMcomb. It consists of three main steps: (a) Construction of the probability model generated from HMM. The parameters of a statistical model representing the family are estimated using the training examples, in conjunction with general a priori information about properties of proteins. (b) Features extraction and representation. The model generated in the previous step assigns a probability to any given protein sequence. (c) Classification step in which we construct SVM classifiers to determine whether the protein belongs to the predefined family or not.

Corresponding Author: Dr. Nazar Zaki, College of IT, UAE University, Al Ain 17555, UAE
Email: nzaki@uaeu.ac.ae, Fax: +971-3-7626309

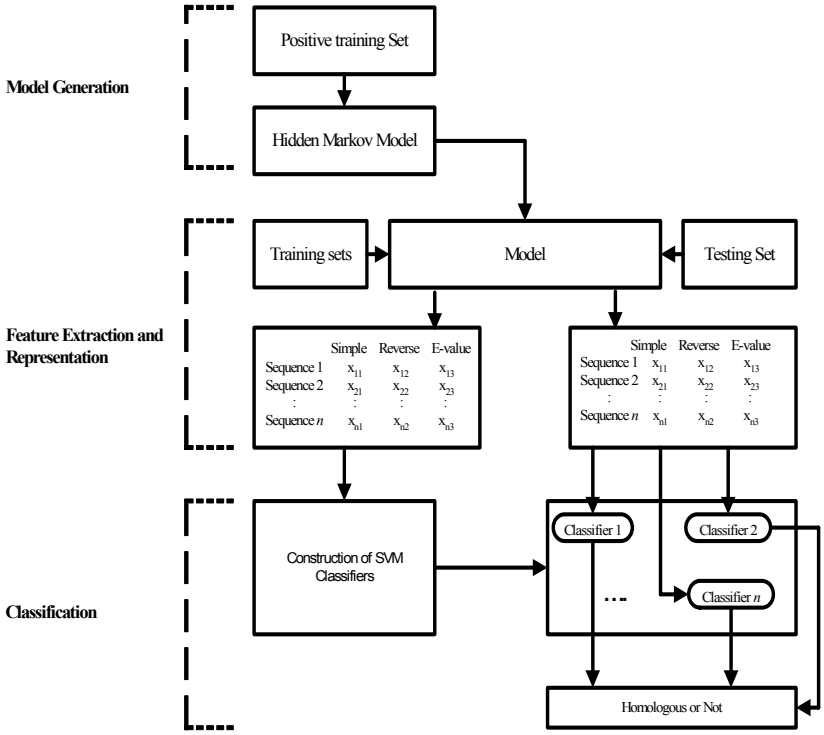

Fig. 1: Overview of the SVM-HMMcomb method.

Construction of the Probability Model: The model is a linear sequence of *nodes*, each of which includes *match*, *insert*, and *delete* states. Each match state has a distribution over the appropriate amino acid indicating which characters are most likely. The chain of match states forms a model of the protein family, or of columns of a multiple alignment. For more details of the construction of the probability model generated from HMM, [6, 8, 9].

Protein Sequence Representation: We combined three different scoring algorithms to extract the maximum biological information in a minimum number of dimensions. The combination, yields greater and absolutely faster computational efficiency. The three scoring algorithms are presented by Barrett et al. [13]:

- Simple: NLL-NULL score which is the negative log-likelihood (NLL) scores, $-\log(P(s|m))$ using a simple null model (NULL) based on the probability that the sequence *s* was generated by the model *m*.
- Reverse: NLL-NULL score for the reverse sequence NULL model
- E-value score

The scoring algorithm used in this study provides several less biased means of scoring by reporting NLL scores as the difference between a null model and trained model NLL score. The scoring program can find NLL and NLL-NULL (log-odds) scores. E-values can be calculated for the reverse sequence null model. The most common operation is to calculate NLL-NULL scores for a large number of sequences. This can be done by supplying the model file and one or more sequence database files. The E-value computation is based on a simple assumption that the scores for the sequence and the reversed-sequence are independent draws from an extreme-value distribution:

$$P(score < a) \approx 1 - e^{(-ke^{\lambda a})} \qquad (1)$$

Subtracting the two scores gives

$$P(diff < a) \approx \frac{1}{1+e^{-\lambda a}} \qquad (2)$$

The E-value is the expected number of sequences with that good score, so is simply the probability of seeing the negative difference, multiplied by the number of sequences scored. The only parameter that needs to be estimated is the natural scaling λ. Since we use natural logs in computing our probabilities, we set $\lambda = 1$, which seems to be correct experimentally. Thus we compute the E-values with no parameter fitting at all. It is based purely on theoretical considerations.

The representation of the protein sequences using the combination of the above scoring algorithms is performed using the Sequence Alignment and Modeling system (SAM) [14].

Construction of SVM Classifiers: The SVM algorithm, addresses the general problem of learning to discriminate

between positive and negative members of a given class of n-dimensional vectors. The algorithm operates by mapping the given training set into a possibly high-dimensional feature space and attempting to learn a separating hyperplane between the positive and the negative examples for possible maximization of the margin between them. The margin corresponds to the distance between the points residing on the two edges of the hyperplane. Having found such a plane, the SVM can then predict the classification of an unlabeled example. In fact, much of the SVM's power comes from its criterion for selecting a separating plane when many candidate planes exist: the SVM chooses the plane that maintains a maximum margin from any point in the training set [15]. Statistical learning theory suggests that, for some classes of well-behaved data, the choice of the maximum margin hyperplane will lead to maximal generalization when predicting the classification of previously unseen examples [10]. SVM classifiers do not require any complex parameters to be tuned and optimized, and they exhibit a great ability to generalize even when given a small number of training examples. The only significant parameters to be tuned are the choice of the kernel function and the soft-margin parameter (capacity, regularization parameter). The soft-margin parameter used to determine the trade-off and allows us to control how much tolerance for errors in the classification of the training samples we should allow. It's therefore, effect the generalization ability of the SVM classifier and prevents it form overfitting the training dataset [16]. The second tuning parameter is the kernel. SVM uses the kernel function to create the hyperplane in high dimensional spaces that effectively separate the training data. Often in the input space training, vectors cannot be separated by a simple hyperplane. The kernel projects the data to higher dimensional space to increase the computational ability. Finding an appropriate kernel function for a particular application area can be difficult and remains largely an unresolved issue [17]. In our implementation, we used gist SVM software implemented by Noble et al. [18]. In all the experiments, the soft-margin parameter is set to 1000 and employed the Gaussian Radial Basis Function kernel (RBF kernel). The Gaussian Radial Basis function is used as it allows pockets of data to be classified which is more powerful way than just using a linear dot produce.

EXPERIMENTS

The performance of our technique is tested on the SCOP database version 1.37 PDB90 [19]. The use of SCOP (Version 1.37), datasets designed by Jaakkola *et al* [9] allows direct comparison with the previous work on protein remote homology detection. He selected for the test all SCOP families that contain at least 5 PDB90 sequences and have at least 10 PDB90 sequences in the other families in their superfamily. This process results in 33 test families from 16 superfamilies. The *positive test examples* are simulated by members of a target SCOP family from a given superfamily. *Positive training* examples are chosen from the remaining families in the same superfamily. While *negative test* and *negative training* examples are chosen from disjoint sets of folds outside the target family's fold [20]. The generative models of each family are also available and they obtained from an existing library of SAM-T98 HMMs.

Evaluation Measures of the Method Performance: The performance of the SVM-HMMcomb method is measured by how well the method can assign novel protein sequence to its correct family. The method can make errors by assigning the sequences to families to which they do not belong or failing to assign the sequences to families to which they do belong. For such a binary classification problem, there are two classes $\{-1,+1\}$ = {unrelated, related}. The positive sequences or the sequences that belong to the family "+1" are considered as related sequences, whereas the negative sequences are the unrelated sequences.

The information encoded in the contingency table is used to calculate the protein homology detection evaluation measures. We used two evaluation measures:

- Rate of False Positive (RFP), which defined as the fraction of negative test sequences that score as high, or better than the positive sequence.

- We further more need to calculate the receiver operating characteristic (ROC) [21] of the SVM-HMMcomb method. The ROC statistic is the integral of the ROC curve, which plots the True Positive Proportion (recall), $tpp = \frac{tp}{(tp + fn)}$, versus the False Positive Proportion (precision), $fpp = \frac{tp}{(tp + fp)}$. The ROC statistic was calculated by scoring all the positions in the test set using the log-odds matrix, sorting the positions by score, and then numerically integrating *tpp* over *fpp* using the trapezoid rule.

RESULTS

In Fig. 2, we illustrate the overall performance of our protein remote homology detection method on the 33 test families. We also show the overall performance of other existing protein remote homology detection methods such SVM-Fisher method as a discriminative model and SAM as a purely generative mode. The performance is measured based on the median RFP. Another comparison between the SVM-Fisher and SVM-HMMcomb is illustrated in Fig. 3 in terms of the maximum RFP. We plot the number of SCOP families with given performance on the X-axis while on the Y-axis we plot values for the median or maximum RFP. It's clear to notice that SVM-HMMcomb achieved stable RFP values for most of the families compared to the other methods (Fig. 2, 3).

We report further experimental work to compare the performance of SVM-HMMcomb approach to the recently

introduced method such as SVM-Mismatch kernel methods [12]. The graph ranks the homology detection methods according to ROC. A higher curve corresponds to more accurate homology detection performance (Fig. 4). From the graph, we observe that SVM-HMMcomb performs better than all other methods.

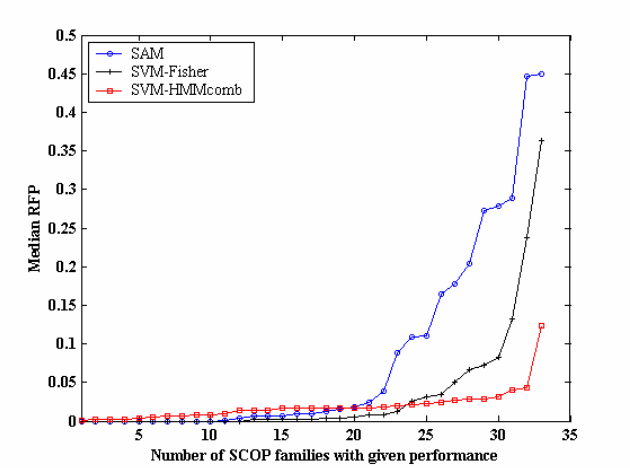

Fig. 2: Median RFP comparison of SVM-HMMcom method with SAM, SVM-Fisher.

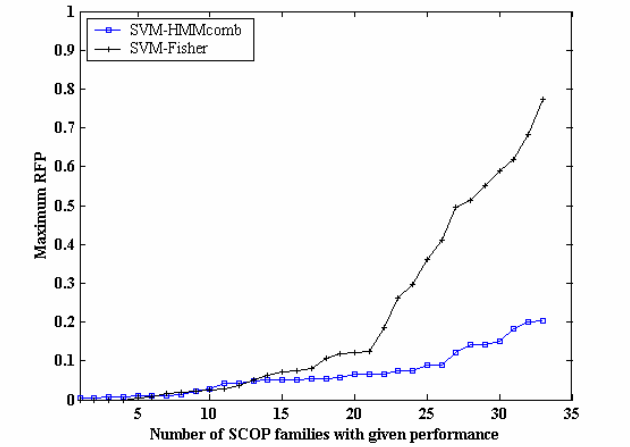

Fig. 3: Maximum RFP comparison of SVM-HMMcom method with SVM-Fisher.

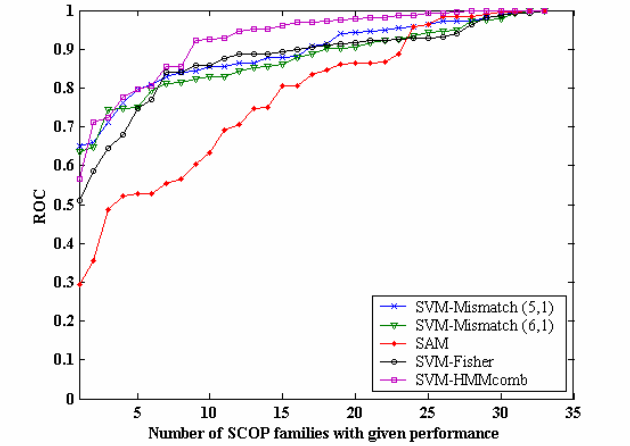

Fig. 4: ROC overall performance comparison.

More detailed comparison is made between our approach and SVM-Fisher, SAM, and SVM-Mismatch (with sub-string length of 5, and 6) in terms of which method achieves better ROC scores. Family-by-family comparison of the SVM-HMMcomb performance against SVM-Mismatch method is shown in Fig. 6a, 6b. Each family is plotted as a point (x, y) where x is the ROC score for the SVM-HMMcomb and y is the ROC score for the SVM-Mismatch method. Another close comparison is made between our approach and SVM-Fisher method. Family-by-family comparison of the SVM-HMMcomb method performance against SVM-Fisher method is illustrated in Fig. 6c. Each family is again plotted as a point (x, y) where x is the ROC score for the SVM-HMMcomb and y is the ROC score for the SVM-Fisher method.

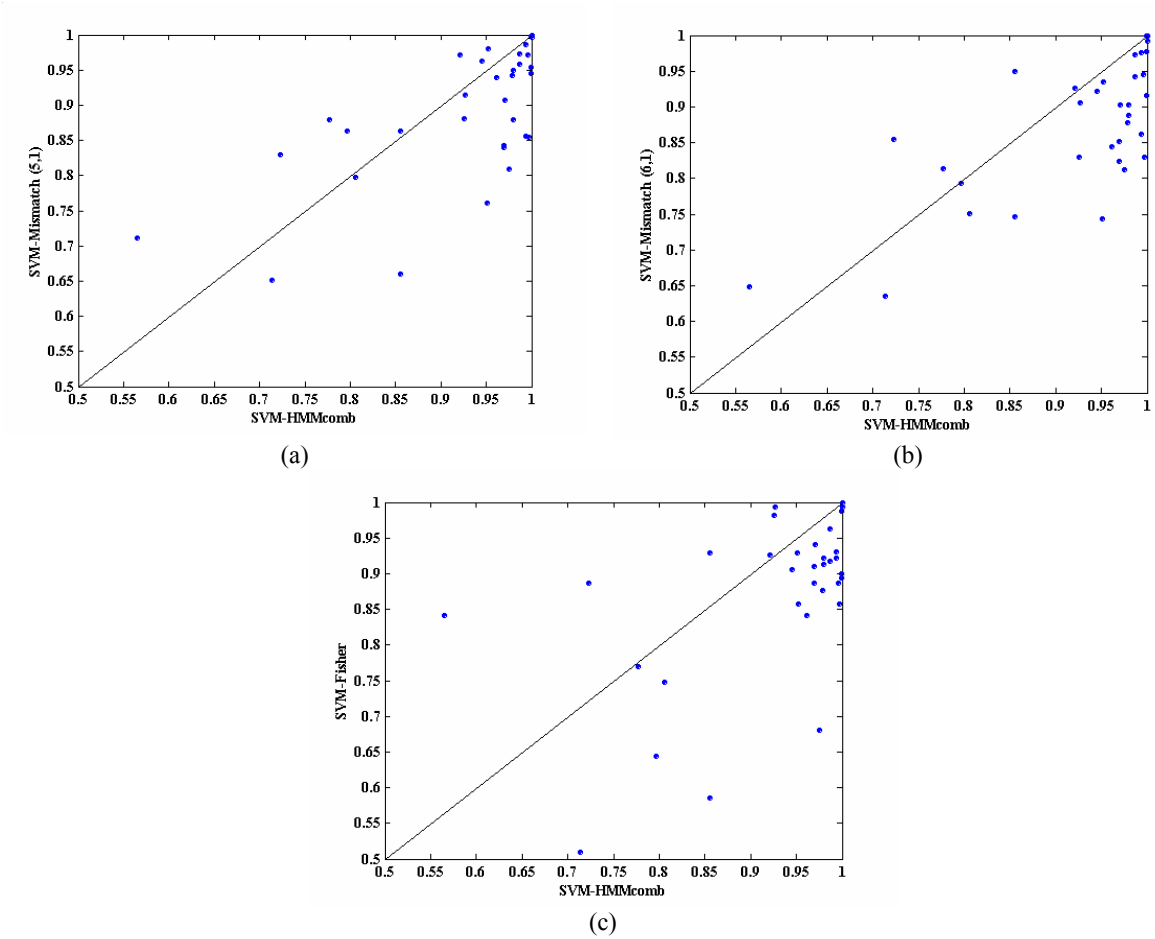

Fig. 5: Family-by-family Comparison.

We assess the statistical significance of the differences among methods using two-tailed signed rank test [22, 23]. We represented the *p*-values from a two sample, 2-tailed *t*-test. This means that the probability of falsely concluding the alternative hypothesis is the value shown. An entry (in bold) in the table indicates that the method listed in the current row performs significantly better than the method listed in the current column. As SVM-HMMcomb method perform significantly better (*p*-value = 0.05) than SAM and SVM-Mismatch with sub-string length 6. Note that a bold value indicates that the *p*-value is less than 0.05 (Table1).

Table 1: Statistical Significance of Differences between Pairs of Homology Detection Methods.

	Mismatch (5,1)	Mismatch (6,1)	SVM-Fisher	SAM
SVM-HMMcomb	0.170742	**0.042294**	0.06113	**0.00033**
Mismatch (5,1)		0.471212	0.49026	**0.00382**
Mismatch (6,1)			0.95855	**0.01240**
SVM-Fisher				**0.01820**

DISCUSSION

The SVM-HMMcomb method introduced in this paper, and the representation of the protein sequence, offers two main advantages over the Fisher and mismatch representations. First, it combines the power of different scoring methods to produce richer representation of the protein sequence. More meaningful features yield better generalization performance. While, for instance, the sufficient statistics used in SVM-Fisher method reflect the summary of the relevant information for the likelihood produced by forward-backward as a single scoring algorithm. Second; SVM-HMMcomb method is simpler, in the sense we need to deal with only three dimensions unlike SVM-Fisher and SVM-Mismatch methods which require more time and memory to compute the kernel matrix.

The idea of combining several algorithms to increase the classification power is not novel [24, 25] however, the novelty of our method is the use of the combination in conjunction with SVM and applying the idea to a sensitive problem such as remote protein homology detection.

One significant characteristic of any protein remote homology detection algorithm is whether the method is computationally efficient or not. In this respect, all SVM-Fisher and SVM-Mismatch algorithms include an SVM optimization, which is roughly $O(n^2)$, where n is the number of training set examples. However, the computation of the kernel matrix in SVM-HMMcomb method is logically much faster and requires less memory since we deal with only three dimensions. The computational cost of the SVM optimization is only $O(n^2)$. For the feature extraction step; both SVM-HMMcomb and SVM-Fisher require $O(nmp)$ running time. Where n is the number of the training examples, m is the length of the longest protein sequence, and p is the number of the HMM parameters. The (k,m) mismatch kernel can be computed in $O(nk^m l^m)$, where k is the length of the amino acid subsequence, m is the mismatch and l is the size of the alphabets. Thus assuming that $k^m \approx p$ and $l^m \approx m$, the SVM-HMMcomb and mismatch kernel methods roughly take the same running time for the computation of the kernel matrix.

One important drawback of the method introduced here is the need for lot of data or prior knowledge to train the hidden Markov model.

CONCLUSION

In this study, we have introduced a simple method for the recognition of the remote protein homology. We combine techniques from machine learning and information extraction to capture the evolutionary relationships between protein sequences. This work is focused on achieving better protein homology detection with minimum fixed-length vectors representation of the protein sequence. This is done through incorporating the maximum biological information and evolutionary relationships into a fix-dimension of the length three. With this simple representation, we are able to achieve superior performance to previously presented protein homology detection methods while achieving better computational cost. However, SVM-HMMcomb algorithm is not significantly better in terms of computational efficiency than SVM-Fisher and SVM-Mismatch. In the future, we plan to investigate the SVM-HMMcomb method performance on a benchmark dataset developed recently by Liao et al. [15]. The dataset has very limited positive training examples and no additional horologes are added. Further analysis of the features extracted is remaining to be investigated. This analysis will make it clear whether all the three types of features are equally important. The possibility of gaining better performance by using additional predictors is yet to be done. It will be important to extend the method to identify multiple domains within large protein dataset.

ACKNOWLEDGEMENTS

The authors thank Nello Cristianini (*University of California, Davis*), Chih.-Jen Lin (*National Taiwan University*) and Christina Leslie (*Columbia University*) for their valuable comments and co-operation.

REFERENCES

1. Smith, T. and M. Waterman, 1981. Identification of common molecular subsequence. *J. Mol. Biol.,* 147: 195-197.
2. Pearson, W. R., 1985. Rapid and sensitive sequence comparisons with FASTAP and FASTA. *Method. Enzymol.,* 183: 63-98.
3. Altschul, S. F., W. Gish, W. Miller, E. Myer and J. Lipman, 1990. Basic local alignment search tool. *J. Mol. Biol.,* 215: 403-410.
4. Gribskov, M., R. Lüthy and D. Eisenberg, 1990. Profile analysis. *Method. Enzymol.*, 183: 146-159.
5. Baldi, P., Y. Chauvin, T. Hunkapiller and M. A. McClure, 1994. Hidden Markov models of biological primary sequence information. *Proc. Nati. Acad. Sci. USA*, 91: 1059-1063.
6. Krogh, A., M. Brown, I. S. Mian, K. Sjölander D. Haussler, 1994. Hidden Markov models in computational biology: Applications to protein modeling. *J. Mol. Biol.*, 235: 1501-1531.
7. Altschul, S. F., T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller and D. J. Lipman, 1997. Gapped Blast and Psi-Blast: a new generation of protein database search programs. *Nuc. Acid. Res.*, 25: 3389-3402.
8. Karplus, K., C. Barrett and R. Hughey, 1998. Hidden Markov models for detecting remote protein homologies. *Bioinformatics*, 14: 846-56.
9. Jaakkola, T., M. Diekhans and D. Haussler, 2000. A discriminative framework for detecting remote protein homologies. *J. Comp. Biol.*, 7: 95-114.
10. Vapnik, V. N., 1998. *Statistical Learning Theory*. New York, USA: Wiley.

11. Cristianini, N., and J. Shawe-Taylor, 2000. *An introduction to Support Vector Machines*. Cambridge, UK: Cambridge University Press.
12. Leslie, C., E. Eskin, J. Weston and W. Noble, 2004. Mismatch String Kernels for Discriminative Protein Classification. *Bioinformatics*, 20: 67-76.
13. Barrett, C., R. Hughey, K. Karplus, 1997. Scoring Hidden Markov Models. *CABIOS,* 13:191-199.
14. SAM software, http://www.cse.ucsc.edu/research/compbio/sam.html.
15. Liao, L. and W. S. Noble, 2003. Combining Pairwise Sequence Similarity and Support Vector Machines for Detecting Remote Protein Evolutionary and Structural Relationships. *J. Comp. Biol.*, 10: 857-868.
16. Logan, B., P. Moreno, B. Suzek, Z. Weng and S. Kasif, 2001. *A Study of Remote Homology Detection*. Tech. Rep., Cambridge Res. Lab.
17. Jaakkola, T., M. Diekhans and D. Haussler, 1999. Using the Fisher kernel method to detect remote protein homologies. *Proc. 7th ISMB*, Menlo Park, CA. AAAI Press, pp: 149-158.
18. GIST software, http://www.cs.columbia.edu/compbio-/svm.
19. Murzin A. G., S. E. Brenner, T. Hubbard and C. Chothia 1995. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.* 247: 536-540.
20. Dataset, http://www.cse.ucsc.edu/research/compbio/discriminative
21. Swets, 1988. Measuring the accuracy of diagnostic systems. *Science*, 270: 1285-1293.
22. Salzberg, S. L., 1997. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining Know. Dis.*, 1: 317-328.
23. Henikoff, S. and J. G. Henikoff, 1997. Embedding strategies for effective use of information from multiple sequence alignments. *Protein Sci.* 6: 698-705.
24. Shimshoni, Y., and N. Intrator, 1998. Classifying seismic signals by integrating ensembles of neural networks, *IEEE Signal Proces.* 46: 1194-1201.
25. Merz, C., 1999. Using correspondence analysis to combine classifiers. *Mac. Learn.* 36: 33-58.