International Journal of Biological, Life and Agricultural Sciences
ISSN: 2415-6612
Vol:2, No:1, 2008

Detecting Remote Protein Evolutionary Relationships via String Scoring Method

Nazar Zaki and Safaai Deris

***Abstract*—**The amount of the information being churned out by the field of biology has jumped manifold and now requires the extensive use of computer techniques for the management of this information. The predominance of biological information such as protein sequence similarity in the biological information sea is key information for detecting protein evolutionary relationship. Protein sequence similarity typically implies homology, which in turn may imply structural and functional similarities. In this work, we propose, a learning method for detecting remote protein homology. The proposed method uses a transformation that converts protein sequence into fixed-dimensional representative feature vectors. Each feature vector records the sensitivity of a protein sequence to a set of amino acids substrings generated from the protein sequences of interest. These features are then used in conjunction with support vector machines for the detection of the protein remote homology. The proposed method is tested and evaluated on two different benchmark protein datasets and it's able to deliver improvements over most of the existing homology detection methods.

***Keywords*—**Protein homology detection; support vector machine; string kernel.

I. Introduction

THE recent years have witnessed a consistent surge in sequence information, caused by technological breakthroughs in large-scale sequencing projects. The main challenge facing biologist now, is to interpret this newly generated sequence data. One way to achieve this goal is through protein homology detection. Much research has already been done in protein homology detection. 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 biologists 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 Support Vector Machine (SVM) [10]-[11], is currently among the well known methods for detecting remote protein homology. Other HMM base method is the HMM Combining Score (HMMcs) method [12]. HMMcs added more improvement over SVM-Fisher; however, both methods are appealing because they combine the rich biological information encoded in a profile HMM with the discriminative power of the SVM classifiers. In this case, we generally need lot of data or prior knowledge to train HMM [13].

Recently, two strings base methods are introduced. The first is the mismatch kernels method [13] and the second is the string kernel method designed by Zaki et al. [14]. In the second method, the authors introduced the application of the string kernel (SK) in classifying protein sequence. The string kernels approach has been shown to achieve good performance on text categorization tasks [15]. The basic idea is to compare two protein sequences by looking at common subsequences of a fix-length. These two methods were able to perform well on classifying protein sequence; however, no biological information is incorporated and the two techniques do not use any domain knowledge. They consider the protein dataset just as a long string of text.

Other known method is the SVM-Pairwise method developed by Liao et al. [16]. The method means of representing proteins using pairwise sequence similarity scores. The drawback of this method is the fact that, when we compute the similarity scores, we consider all the sequence. It will be more meaningful if we could split the sequence into substrings and then measure the similarity score based on sensitive and non-sensitive regions.

In this paper, we combined the advantage of using string kernel and incorporating some biological knowledge by using SVM-Pairwise concepts. The method uses a transformation that converts protein sequence into fixed-dimensional representative feature vectors where each feature vector records the sensitivity of a protein sequence to a set of amino acids substrings generated from the protein sequences of interest. The method is called SVM String Scoring (SVM-SS) method.

This work was financially supported by the Research Affairs at the UAE University under a contract no. 05-01-9-11/05.

Nazar Zaki is an Assistant Professor with the College of Information Technology, United Arab Emirates University (UAEU). Al-Ain 17555 UAE, (phone: +971-50-7332135; fax: +971-3-7626309; e-mail: nzaki@uaeu.ac.ae).

Safaai Deris is a Professor with the Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia, (e-mail: safaai@fsksm.utm.my).

II. SVM-SS OVERVIEW

SVM-SS method for detecting remote protein homology consists of two main steps: First, converting all the protein sequences of interest into high dimensional feature vectors. We create each vector by scoring a set of substrings against each protein sequence. Once this transformation has taken place, we then compute the kernel matrix to be used in conjunction with SVM. SVM discriminators will then separate each protein family from the rest. We show this process in Figure 1. The description of each step is given in the following sections.

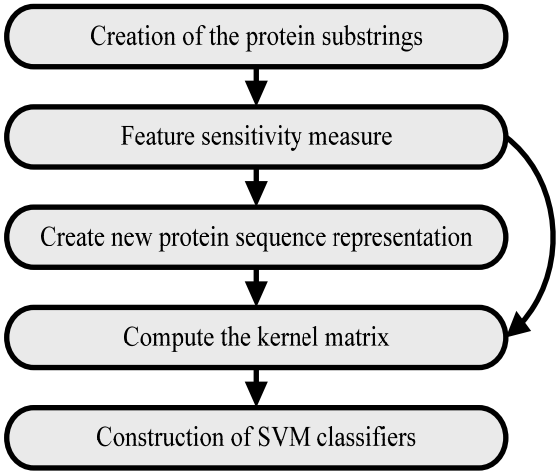


Fig. 1 Overall SVM-SS method

A. Creation of the Protein Substrings

In this step, we consider each protein sequence as a string of amino acids and then, we try to find out all possible substrings that the protein sequence contains. Unlike the string kernel method [14], we consider only the contiguous substrings. This goal can easily be achieved by simply shifting a window of a length $k > 1$, over the protein training examples. This process can be illustrated as follows:

If we have a protein sequence

>e1izb.2c 7.1.1.1.3 Insulin {Pig (Sus scrofa)}
giveqcctsicslyqlenycn
Assume $k = 5$, yields 4 substrings (in bold).

giveqcctsicslyqlenyc	**giveq**
giveq**cctsi**cslyqlenyc →	**cctsi**
giveqcctsi**cslyq**lenyc	**cslyq**
giveqcctsicslyq**lenyc**	**lenyc**

B. Feature Sensitivity Measure

To create a feature vector for each protein sequence, we have to search for each substring in the protein dataset. This will result in *n*-dimensional feature vector where *n* is the total number of substrings extracted from the protein sequences. All the *n* substrings are then scored against the protein sequences of interest. This vectorization step uses the Smith-Waterman algorithm as implemented in Fasta [2]. The feature vector corresponding to protein *m* is $F_m = f_{m1}, f_{m2}, ..., f_{mn}$, where *n* is the total number of the substrings generated, and f_{mi} is the E-value of the Smith-Waterman score between sequence *m* and the i^{th} substring. The process is shown in Figure 2.

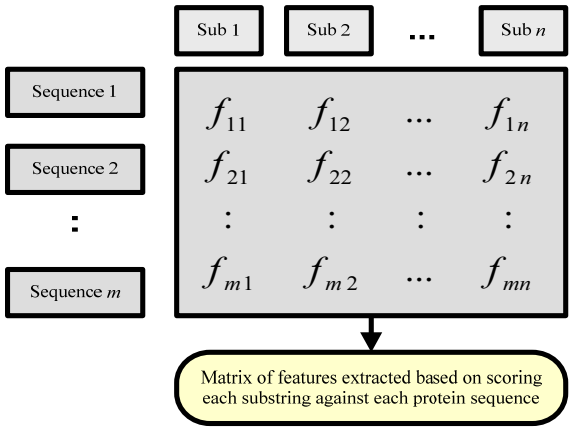


Fig. 2 The procedure to generate feature vectors from protein sequences

C. Construction of SVM Classifiers

SVM is a powerful classification algorithm and well suited the given task. It 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 [12]. This margin roughly corresponds to the distance between the points residing on the edges of the hyperplane [17]. Having found such a plane, the SVM can then predict the classification of an unlabeled example. The formulation of the SVM is described as follows:

Suppose our training set S consists of labeled input vectors (x_i, y_i), $i = 1...m$ where $x_i \in \Re^n$ and $y_i \in \{\pm 1\}$. We can specify a linear classification rule f by a pair (w, b), where the normal vector $w \in \Re^n$ and the bias $b \in \Re$, via

$$f(x) = (w, b) + b \quad (1)$$

where a point x is classified as positive if $f(x) > 0$. Geometrically, the decision boundary is the hyperplane

$$\{x \in \Re^n : (w, x) + b = 0\} \quad (2)$$

The idea makes it possible to efficiently deal with vary high dimensional futures spaces is the use of kernels:

$$K(x, z) = \langle \phi(x) \cdot \phi(z) \rangle \quad \text{for all } x, z \in X \quad (3)$$

where ϕ is the mapping from X to an inner product feature space. We thus get the following optimization problem:

$$\max_{\lambda} \sum_{i=1}^{m} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{m} \lambda_i \lambda_j y_i y_j K(x_i, x_j) \quad (4)$$

subject to the constraints

$$\lambda_i \geq 0 \qquad \sum_{i=1}^{m} \lambda_i y_i = 0 \quad (5)$$

The appeal of SVMs is twofold; first they do not require any complex tuning of parameters, and second they exhibit a great ability to generalize given small training samples. In this particular implementation, we used the Gist SVM software available at http://www.cs.columbia.edu/compbio-/svm.

III. EXPERIMENTAL WORK

In this section, we report the experimental work done to apply the SVM-SS method for detecting protein homology. We first converted all protein sequences of interest into high dimensional feature vectors. We created each vector by scoring a set of the generated substrings against the protein sequences of the interest. Therefore, in order to create feature vectors, we first need to create a database of substrings. In this case, we shifted a window of a length equal to 14 (we have tried different length, results not shown). To represent a protein sequence by feature vectors one has to search for each substring in this protein sequence. Each substring scores against the protein sequence using pairwaise scoring algorithm. Only default parameters are used: gap opening penalty and extension penalties of 11 and 1, respectively, and the position specific scoring matrix BLOSUM 62. The result of this transformation is an *n*-dimensional feature vector, where *n* is the total number of substrings created. We also applied a threshold value in such a way that, scores below the threshold are set to zero. In our case, the threshold value was set to 1. Once the transformation of the protein sequences is done, we then learn SVM discriminators to separate protein families. We primarily employ the Gaussian kernel for all classifiers. In terms of soft-margin parameter, we choose a value that is close to the absolute maximum kernel distance, in our case the value is 100. This choice of capacity guarantees the numerical stability of the SVM algorithm and provides sufficient generalization. This solution is a clean way to set the tuned parameters solely based on the training set.

A. Protein Data Used

The performance of the SVM-SS algorithm is tested on two SCOP [18] benchmarked databases, the first is designed by Jaakkola et al [9] which allow 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. Details about this dataset are available at www.cse.ucsc.edu/research/compbio-/discriminative. The second benchmarked database is the SCOP version 1.53. Sequences were selected using the Astral database [19], removing similar sequences using an E-value threshold of 10^{-25}. This procedure yields 4352 distinct protein sequences, grouped into families and superfamilies. For each family, the protein domains within the family are considered positive test examples, and the protein domains outside the family but within the same superfamily are taken as positive training examples. The data set results in 54 families containing at least 10 family members and 5 superfamily members outside of the family. Negative examples are taken from outside of the positive sequences fold, and are randomly split into train and test sets in the same ratio as the positive examples. Details about the various families and the complete data set are available at: http://www.cs.columbia.edu/-compbio/svm-pairwise.

B. Comparing SVM-SS with the known homology detection methods

The performance of the SVM-SS method is then compared to eight of the well known methods such as: HMMER [20], BLAST [3], SAM [8], SVM-Fisher [9], Mismatch String Kernels [13], SVM-Pairwise [16], HMMcs [12], and SVM-SK [14].

C. Evaluation Measures of the Method Performance

The performance of the SVM-SS 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. To that end, two methods are used to evaluate the performance of the protein homology detection:

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

- Receiver Operating Characteristic (ROC) [21] of the SVM-SS 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), $fpp = \frac{tp}{(tp + fp)}$.

  Where *tp* is the true positive, *fn* is the false negative, and *fp* is the false positive.

IV. Results and Analysis

The performances of the homology detection algorithms are measured by how well these algorithms can assign novel protein sequence to its correct family. The performance of the proposed algorithm is compared with the current successful homology detection methods on a benchmark datasets from SCOP database. We first tested the performance of SVM-SS method using the SCOP (Version 1.37) data sets. In Figure 3, we illustrate the relative overall performance of the SVM-SS algorithm on the 33 test families. The figure also shows the overall performance of other existing protein remote homology detection methods. The methods included are HMMER, BLAST, and SAM as a purely generative models and SVM-Fisher as a combination of the generative and discriminative models. We also included HMMcs and SVM-SK methods. The graph ranked the seven homology detection methods according to their Rate of False Positives (RFP) scores. In each graph a lower curve corresponds to more accurate homology detection performance. Using the RFP performance measure, the SVM-SS performs slightly better than the other six methods including SVM-Fisher, HMMcs and SVM-SK Methods.

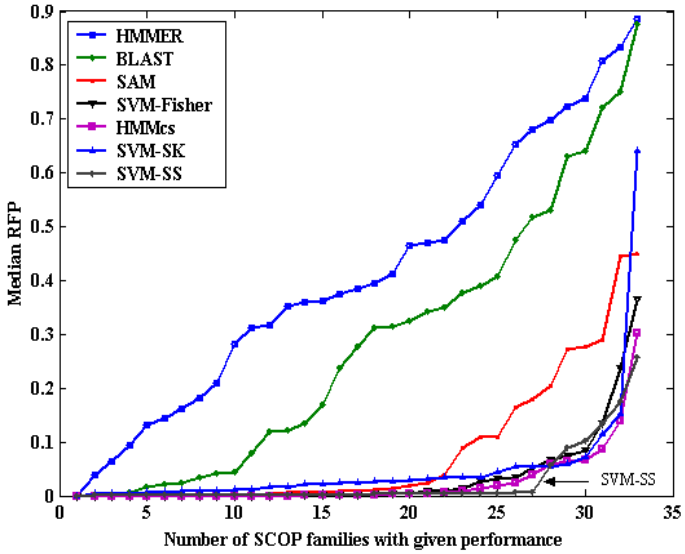
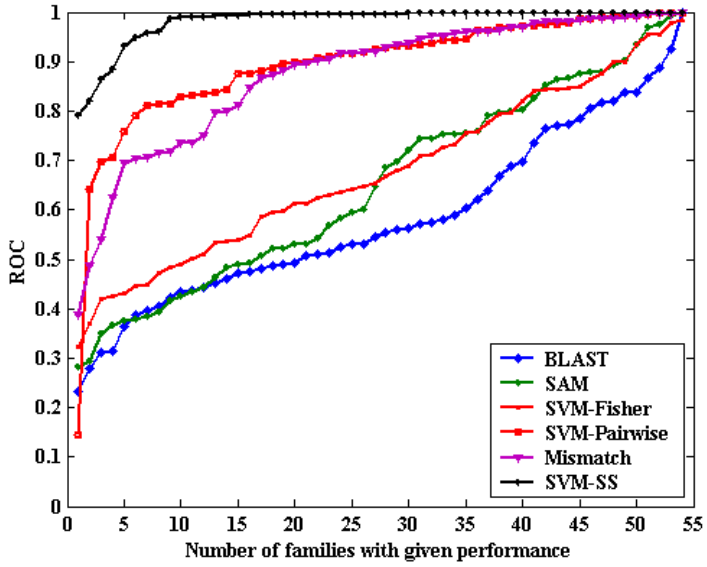


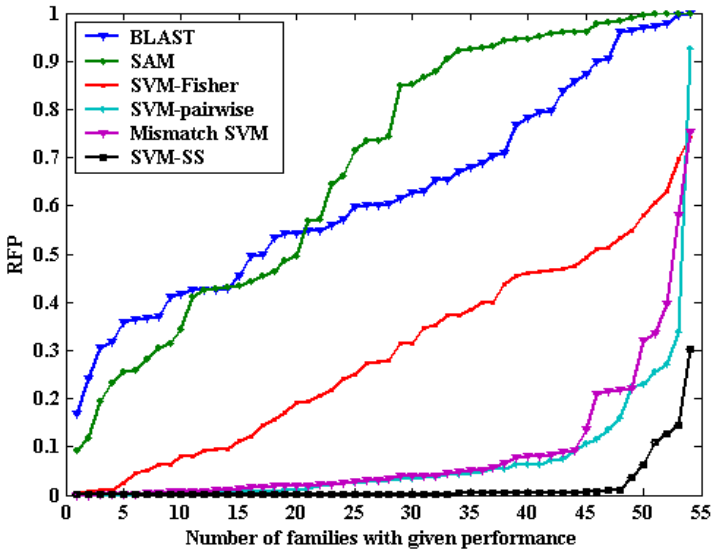
Fig. 3 Overall performances in terms of RFP of protein homology detection methods on the 33 SCOP test families

In this section, we report further experimental work to compare the performance of SVM-SS to the recently introduced methods such as SVM-Paiwise and Mismatch kernel methods. We tested the performance of our algorithm on the SCOP database version 1.53. The use of SCOP version 1.53 [16], allows direct comparison with the previous work on remote homology detection. We included in the comparison, BLAST, SAM, SVM-Fisher, SVM-Pairwise and Mismatch kernel methods.

The results of the comparatives experiment are summarized in Figure 4 (a) and (b). The two graphs rank the six homology detection methods according to ROC and median RFP scores. In Figure 4 (a), a higher curve corresponds to more accurate homology detection performance. While in Figure 4 (b), a lower curve corresponds to more accurate homology detection performance. From the two graphs, we observe that SVM-SS performs significantly better than all other methods.



(a)



(b)

Fig. 4 Relative performance of homology detection methods using (a) ROC (b) RFP

In Figure 5, Figure 6 and Figure 7, we show family-by-family comparison of SVM-SS against SVM-Fisher, SVM-Pairwaise and Mismatch kernel methods in terms of ROC and RFP scores. Each point on the graph corresponds to one of the 54 SCOP families. The axes are ROC (left figure) and RFP (right figure) achieved by the two primary methods compared in this study.

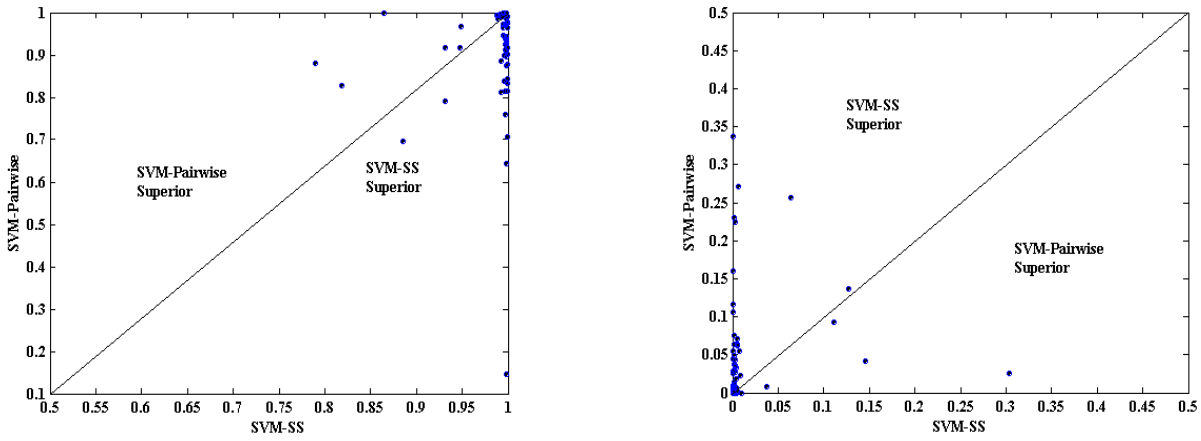


Fig. 5 Family-by-family comparisons of SVM-SS and SVM-Pairwise methods

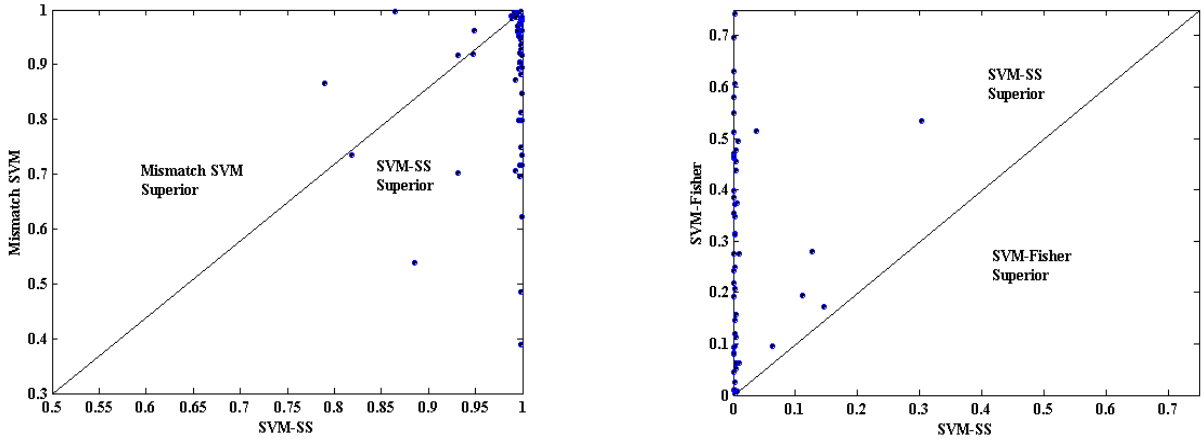


Fig. 6 Family-by-family comparisons of SVM-SS and Mismatch-SVM methods

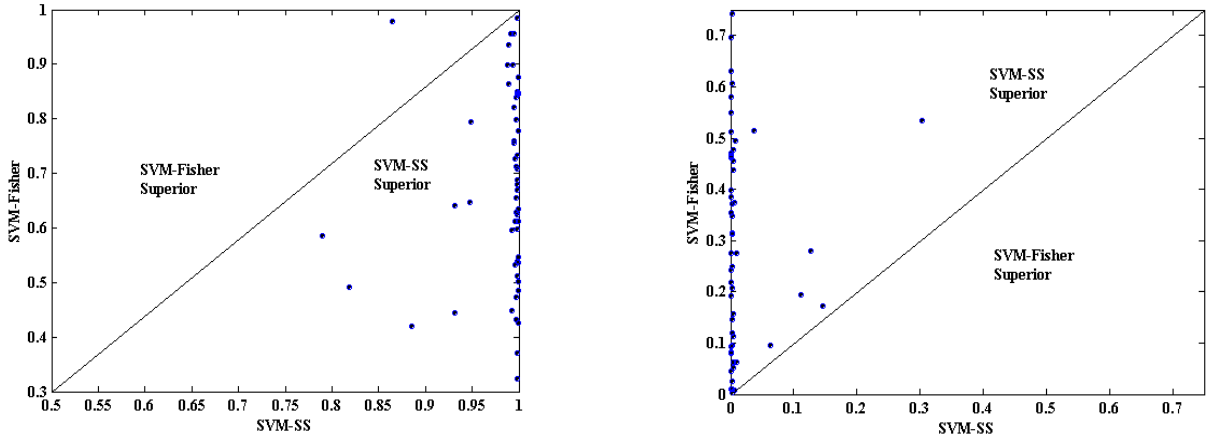


Fig. 7 Family-by-family comparisons of SVM-SS and SVM-Fisher methods

Using either performance measure, the SVM-SS method performs significantly better than the other five methods. We assess the statistical significance of differences among methods using two-tailed signed rank test. As shown in Table

1 (based on ROC scores) and Table 2 (based on RFP scores), nearly all of the differences apparent in Figure 4 and Figure 5 are statistically significant at a threshold of 0.05. The resulting induced performance ranking of methods is SVM-SS, Mismatch-SVM, SVM-Pairwise, SVM-Fisher, SAM and BLAST. Difference between SVM-Pairwise and Mismatch kernel method is however not statistically significant.

TABLE I
STATISTICAL SIGNIFICANCE OF DIFFERENCES BETWEEN PAIRS OF HOMOLOGY DETECTION METHODS BASED ON THE ROC SCORES

Method	Mismatch	Pairwise	Fisher	SAM	Blast
SVM-SS	**0.0**	**0.000012**	**0.0**	**0.0**	**0.0**
Mismatch		0.4486	**0.0**	**0.0**	**0.0**
Pairwise			**0.0**	**0.0**	**0.0**
Fisher				**0.0**	**0.0**
SAM					0.465

TABLE II
STATISTICAL SIGNIFICANCE OF DIFFERENCES BETWEEN PAIRS OF HOMOLOGY DETECTION METHODS BASED ON THE RFP SCORES

Method	Mismatch	Pairwise	Fisher	SAM	Blast
SVM-SS	**0.00106**	**0.00767**	**0.0**	**0.0**	**0.0**
Mismatch		0.57460	**0.0**	**0.0**	**0.0**
Pairwise			**0.0**	**0.0**	**0.0**
Fisher				**0.0**	**0.0**
SAM					0.239

I. CONCLUSION

In this paper, we presented, applied, and analyzed an effective learning method for detecting remote protein homology. The proposed method uses a discriminative framework and in particular Support Vector Machine. It uses a transformation that converts protein domains into fixed-dimensional representative feature vectors, where each feature vector records the sensitivity of a set of substrings to the protein domain of interest. The performance is then tested and evaluated on two different SCOP benchmark protein datasets. The proposed method which we call it SVM-SS method was able to deliver reasonable improvements over most of the existing homology detection methods.

One significant characteristic of any protein remote homology detection algorithm is whether the method is computationally efficient or not. In order to gauge the computational cost of the proposed approach, SVM-SS method is not significantly better than SVM-Fisher, HMMcs, and SVM-SK methods. SVM-SS also includes an SVM optimization, which is roughly $O(n^2)$, where n is the number of training set examples. The feature extraction and representation step of SVM-SS involves computing n^2 pairwise scores. Using Smith-Waterman, each computation is $O(m^2)$, where m is the length of the longest training set sequence, yielding a total running time of $O(n^2m^2)$.

The success of applying the SVM-SS method on detecting protein homology encouraged us to plan future directions such as optimizing the substring width and threshold parameters.

ACKNOWLEDGMENT

This work was financially supported by the Research Affairs at the UAE University under a contract no. 05-01-9-11/05. The authors would also like to acknowledge the help provided by the AI & Bioinformatics Lab (AIBIL) at the Faculty of computer science and information system, Universiti Teknologi Malaysia (UTM).

REFERENCES

[1] T. Smith, and M. Waterman, "Identification of common molecular subsequence", J. Mol. Biol, 147, pp.195, 1981.
[2] W. R. Pearson, "Rapid and sensitive sequence comparisons with FASTAP and FASTA Method", Enzymol, 183, pp. 63, 1985.
[3] S. F. Altschul, W. Gish, W. Miller, E. Myer and J. Lipman "Basic local alignment search tool", J. Mol. Biol., 215, pp. 403, 1990.
[4] M. Gribskov, R. Lüthy and D. Eisenberg, "Profile analysis. Method", Enzymol., 183, pp. 146, 1990.
[5] P. Baldi, Y. Chauvin, T. Hunkapiller and M. A. McClure, "Hidden Markov models of biological primary sequence information", Proc. Nati. Acad. Sci., 91: pp. 1059, 1994.
[6] A. Krogh, M. Brown, I. S. Mian, K. Sjölander D. Haussler, "Hidden Markov models in computational biology: Applications to protein modeling", J. Mol. Biol., 235, pp. 1501, 1994.
[7] 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", Nuc. Acid. Res., 25: pp. 3389, 1997.
[8] K. Karplus, C. Barrett and R. Hughey, "Hidden Markov models for detecting remote protein homologies", Bioinformatics, 14, pp. 846, 1998.
[9] T. Jaakkola, M. Diekhans and D. Haussler "A discriminative framework for detecting remote protein homologies", J. Comp. Biol., 7, pp. 95, 2000.
[10] V. N. Vapnik, "Statistical Learning Theory", John Wiley & Sons, Inc., 1998.
[11] N. Cristianini, and J. Shawe-Taylor, "An introduction to Support Vector Machines", Cambridge, UK: Cambridge University Press. 2000.
[12] N. M. Zaki, S. Deris, and R. M. Illias, "Feature Extraction for Protein Homologies Detection Using Markov Models Combining Scores", Int. J. on Comp. Intelligence and Appl., 1, pp. 1, 2004.
[13] C. Leslie, E. Eskin, J. Weston and W. Noble, "Mismatch String Kernels for Discriminative Protein Classification", Bioinformatics, 20, pp. 67, 2004.
[14] N. M. Zaki, S. Deris, and R. M. Illias, "Application of string kernels in protein sequence classification", App. Bioinformatics, 1, pp. 45, 2005.
[15] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text Classification using String Kernels", J. Machine Learning Res., 2, pp. 419, 2002.
[16] L. Liao, and W. S. Noble, "Combining Pairwise Sequence Similarity and Support Vector Machines for Detecting Remote Protein Evolutionary and Structural Relationships", J. Comp. Biol., 10, pp. 857, 2003.
[17] Zaki, N. M. and Deris, S. (2005). "Representing Protein Sequence with Low Number of Dimensions". Journal of Biological Sciences, 5(6): 795-800.
[18] A. G. Murzin, S. E. Brenner T. Hubbard C. Chothia, "SCOP: a structural classification of proteins database for the investigation of sequences and structures", J. Molec. Biol., 247, pp. 536, 1995.
[19] S. E. Brenner, P. Koehl and M. Levitt, "The ASTRAL compendium for sequence and structure analysis", Nucl. Acids Res., 28, pp. 254, 2000.
[20] S. R. Eddy, "Multiple alignment using hidden Markov models," In Proc. of the 3rd ISMB, pp. 114, 1995.
[21] Swets, "Measuring the accuracy of diagnostic systems". Science, 270: 1285-1293. 1988.

Nazar Zaki is is an Assistant Professor with the College of Information Technology, United Arab Emirates University. Al-Ain 17555 UAE. He holds a Ph.D. in Computer Science from Universiti Teknologi Malaysia (UTM), Malaysia, M.sc in Operations Research and B.sc (honor) in Statistics from

Aligarh Muslims University (AMU), India. He received Dean's awards of the best Ph.D. student (UTM) and top M.sc student (AMU), respectively. His research interest includes Bioinformatics, AI, Machine learning, Pattern Recognition, Kernel functions and Support Vector Machine. Dr. Zaki has published many scientific papers in world class journals and conferences. He is a reviewer in many reputed international journals such as Bioinformatics, Oxford Press. Dr. Zaki is a member of the International Federation of Operational Research Societies, International Computing Society, and Asia and Pacific Bioinformatics Net. Dr. Zaki is one of the members who established the Middle East Association of Healthcare Informatics (MEAHI) in early 2005.

Safaai Deris is a Professor of Artificial Intelligence and Software Engineering at the Faculty of Computer Science and Information Systems, Deputy Dean at the School of Graduate Studies, and Director of Laboratory of Artificial Intelligence and Bioinformatics at the Universiti Teknologi Malaysia. He received the MEng degree in Industrial Engineering, and the DEng degree in Computer and System Sciences, both from the Osaka Prefecture University, Japan, in 1989 and 1997 respectively. His recent academic interests include the application and development of intelligent techniques in planning, scheduling, and bioinformatics. deleted from the biography.