International Journal of Computational Intelligence and Applications
Vol. 4, No. 1 (2004) 1–12
© Imperial College Press
Imperial College Press
www.icpress.co.uk

FEATURES EXTRACTION FOR PROTEIN HOMOLOGY DETECTION USING HIDDEN MARKOV MODELS COMBINING SCORES

NAZAR M. ZAKI* and SAFAAI DERIS†

Department of Software Engineering, FSKSM, Universiti Teknologi Malaysia, Malaysia
Skudai 81300, Johor, Malaysia
**nazar@siswa.utm.my*
†*safaai@fsksm.utm.my*

ROSLI M. ILLIAS

Department of Bioprocess Engineering, FKKKSA, Universiti Teknologi Malaysia, Malaysia
Skudai 81300, Johor, Malaysia
i.rosli@fkkksa.utm.my

Received 12 March 2003
Revised 25 November 2003

Few years back, Jaakkola and Haussler published a method of combining generative and discriminative approaches for detecting protein homologies. The method was a variant of support vector machines using a new kernel function called *Fisher Kernel*. They begin by training a generative hidden Markov model for a protein family. Then, using the model, they derive a vector of features called *Fisher scores* that are assigned to the sequence and then use support vector machine in conjunction with the fisher scores for protein homologies detection. In this paper, we revisit the idea of using a discriminative approach, and in particular support vector machines for protein homologies detection. However, in place of the Fisher scoring method, we present a new Hidden Markov Model Combining Scores approach. Six scoring algorithms are combined as a way of extracting features from a protein sequence. Experiments show that our method, improves on previous methods for homologies detection of protein domains.

Keywords: Protein homology detection; hidden Markov models; support vector machines.

1. Introduction

Protein remote homology detection is an important problem in molecular biology, and it has long been a goal for scientists and researchers. Homology is more general term that indicates evolutionary relatedness among protein sequences. Two sequences are said to be homologies if they are both derived from a common ancestral sequence. The terms similarity and homology are often used interchangeable to describe sequences, but, strictly speaking, they mean different issues. Similarity refers to the presence of identical and similar sites in the two sequences, while homology reflects a stronger claim that the two sequences share a common ancestor.[1]

Most of the current genetic annotation systems rely on computational solutions for homologies modeling via sequences or structural similarities. Many statistical, sequence-base approaches have been developed for protein classification and homology detection. These including methods based on pairwise similarity between protein sequences such as Smith–Waterman dynamic programming algorithm[2] and its approximations such as FASTA[3] and BLAST.[4] Further accuracy was achieved by method based on collecting aggregate statistics, such as Profile[5] and Hidden Markov Models (HMMs).[6,7] Additional accuracy was gained by using iterative methods such as PSI-BLAST[8] and SAM-T98.[9] Additional accuracy was gleaned by modeling the difference between positive and negative examples. Explicitly modeling the difference between these two sets of sequences yields an extremely powerful method. The SVM-Fisher method,[10,11] which combined an iterative HMM training scheme with a discriminative algorithm known as *Support Vector Machines* (SVMs),[12,13] is currently among the most accurate known methods for detecting remote protein homology.

SVM-Fisher begins by training a generative Hidden Markov Model (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*. Support Vector Machines (SVMs) in conjunction with the Fisher scores are then used to discriminate between protein families.

In this paper, we revisit the idea of using a discriminative framework for protein homology detection. However, in place of the Fisher scores, we present a new representation of the protein sequence which we call as HMMs combining scores (HMMcs) approach. The new method performs essentially the same calculation done in the SVM-Fisher, but uses only the total probability score and the Viterbi score, computed with respect to three different background models. Six different scoring algorithms are combined to extract features from protein sequences of interest. Both the SVM-Fisher and HMMcs methods convert a given set of protein sequences into fixed-length vectors and then use the classification ability of the SVMs to discriminate between protein families.

The HMMcs representation of the protein sequences offers two main advantages over the Fisher scores. 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, the sufficient statistics used in SVM-Fisher reflect the summary of the relevant information for the likelihood produced by forward-backward as a single scoring algorithm. Second; HMMcs method is simpler, in the sense we need to deal with only six dimensions unlike SVM-Fisher which deals with thousands of dimensions which requires more time and memory to compute the Fisher kernel matrix.

The idea of combining several algorithms to increase the classification power is not novel.[14–16] 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.

2. Hidden Markov Models

A hidden Markov model is a statistical model, which is very well suited for many tasks in molecular biology, although they have been mostly developed for speech recognition since the early 1970s.[17] The HMM generates a protein sequence by *emitting* amino acids as it progresses through a series of states. Each state has a table of amino acid *emission probabilities* similar to those described in a profile model.[5] There are also *transition probabilities* for moving from state to another. By using a dynamic programming method one can generate a multiple alignment of the unaligned sequences from which the model was built. Thus one can inspect the regions in these training sequences that the process is found to be *homologous.* By studying the model itself, one can glean further insight by noting that it reveals about the common structure underlying the sequences in the family. Finally, the model can be used to discriminate between family and non-family sequences when employed for the database searching.[18]

2.1. *Computing sequence likelihoods*

The likelihood of a sequence is computed by using a dynamic programming procedure called the *forward algorithm.* For all HMM states and starting at time 0, the forward algorithm recursively computes the probability of being in state i at time t, when the tth amino acid of the protein sequence is generated. In a real model, many different state paths through a model can generate the same sequence. Therefore, the correct probability of a sequence is the sum of probabilities over all of the possible state paths. Unfortunately, a brute force calculation of this problem is computationally unfeasible, except in the case of very short sequences. Two good alternatives are to calculate the sum over all paths inductively using the *forward algorithm*, or to calculate the most probable path through the model using the *Viterbi algorithm.* Both algorithms are described below.

2.1.1. *The forward recursion* (*Full likelihood score*)

Using the forward algorithm, the new parameter estimates can be calculated in time proportional to the number of states in the model multiplied by the total length of all the training sequences.

Given a protein sequence $X = \{x_1, x_2, \ldots, x_T\}$ and a state sequence $Q = \{q_1, \ldots, q_T\}$ (of the same length) determined from a HMM with parameters Θ, the likelihood of the protein X along the path Q is equal to:

$$p(X|Q,\Theta) = \prod_{i=1}^{T} p(x_i|q_i,\Theta)\,. \tag{1}$$

The likelihood of a protein sequence $X = \{x_1, x_2, \ldots, x_T\}$ with respect to a Hidden Markov Model with parameters Θ expands as follows:

$$p(X|\Theta) = \sum_{\text{every possible } Q} p(X, Q|\Theta) \tag{2}$$

i.e. it is the sum of the joint likelihoods of the sequence over all possible state sequence allowed by the model.

In practice, the enumeration of every possible state sequence is infeasible. Nevertheless, $p(X|\Theta)$ can be computed in a recursive way by the *forward recursion.* This algorithm defines a forward variable $\alpha_t(i)$ corresponding to:

$$\alpha_t(i) = p(x_1, x_2, \ldots, x_t, q^t = q_i|\Theta) \tag{3}$$

i.e. $\alpha_t(i)$ is the probability of having observed the partial sequence $\{x_1, x_2, \ldots, x_t\}$ and being in the state i at time t (event denoted q_i^t in the course), given the parameters Θ.

2.1.2. *The Viterbi algorithm*

In protein remote homology detection, it is useful to associate "*an optimal*" sequence of states to the protein sequence, given the parameters of a model. For instance, in the case of protein homology detection, knowing which frames of features "*belong*" to which state allows locate the motif boundaries across time. This is called the *alignment* of acoustic feature sequences. A reasonable optimality criterion consists in choosing the state sequence (or *path*) that brings a maximum likelihood with respect to a given model. This sequence can be determined recursively via the *Viterbi algorithm.* This algorithm makes use of two variables:

The highest likelihood $\delta_t(i)$ along a single path among all the paths ending in state i at time t:

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} p(q_1, q_2, \ldots, q_{t-1}, q^t = q_i, x_1, x_2, \ldots x_t|\Theta) \tag{4}$$

a variable $\Psi_t(i)$ which allows to keep track of the best path ending in state i at time t:

$$\Psi_t(i) = \arg\max_{q_1, q_2, \ldots, q_{t-1}} p(q_1, q_2, \ldots, q_{t-1}, q^t = q_i, x_1, x_2, \ldots x_t|\Theta)\,. \tag{5}$$

Note that these variables are vectors of (N-2) elements, (N-2) being the number of emitting states.

A multiple alignment can be generated by using the Viterbi algorithm to find the most likely path through the HMM for each sequence. Each match state in the HMM corresponds to a column in the multiple alignment.

3. Experiments

Our algorithm for homology detection consists of two major steps. First we represent a protein sequence as a fixed-length feature vectors (length six) using different scoring algorithm. This can be done by training HMM on the protein family of interest and then construct a generative probability model. Sequences from that family including sequences that were not used as training examples, expect to yield higher score than those outside the family. The resulting vectors are then be used as an input to SVM discriminators to separate each protein family from the rest.

3.1. *Feature vectors generation*

The first step of our procedures is to get the likelihood scores based on the two scoring algorithms described in Sec. 2 (Full likelihood score and Viterbi score). Full likelihood score is the negative log likelihood of the sequence while Viterbi score is the negative log likelihood of the most probable HMM path associated with the sequence. Each of the two methods is generated with respect to the following background model options which end in six scoring methods:

- None (no background model).
- Sequences (find a sequence of parameters which increase the likelihood at each step, and class of models. The Sequences option uses the average composition of the current sequence file as background model).
- Uniform distribution model.

Each scoring method is contributed as one column in the SVM input matrix as shown in Fig. 1. The score corresponds to an alignment of the sequence to the model. In our method we use the global scores corresponding to global alignments where the entire sequence is aligned to the entire HMM. This means computation of the score begins at the first amino acid in the sequence and ends at the last one. When a background model is used, the score is the log odds ratio, that is the log of the ratio of the corresponding probabilities according to the HMM and the background model. The Uniform option corresponds to a uniform distribution model. The sequences option uses the average composition of the current sequence file as background model.

HMMpro[a] was used to generate the scores. HMMpro is a general purpose HMM simulator for the modeling, analysis, classification, and alignment of biological sequences. Given a set of sequences, we estimated HMM parameters by Maximum Likelihood (ML) using Expectation-Maximization (EM). A trained HMM is used to assign a score to any sequence, fragments included.

3.2. *Construction of SVM classifiers*

The idea of the SVM algorithm[12,13] is to map the given training set into a possibly high-dimensional feature space and attempting to locate in that space a hyperplane that maximizes the distance separating the positive from the negative examples. Having found such a hyperplane, the SVM can then predict the classification of an unlabeled example. Much of the SVM's power comes from its criterion for selecting a separating hyperplane that maintains a maximum margin from any point in

[a] http://www.netid.com/index.html

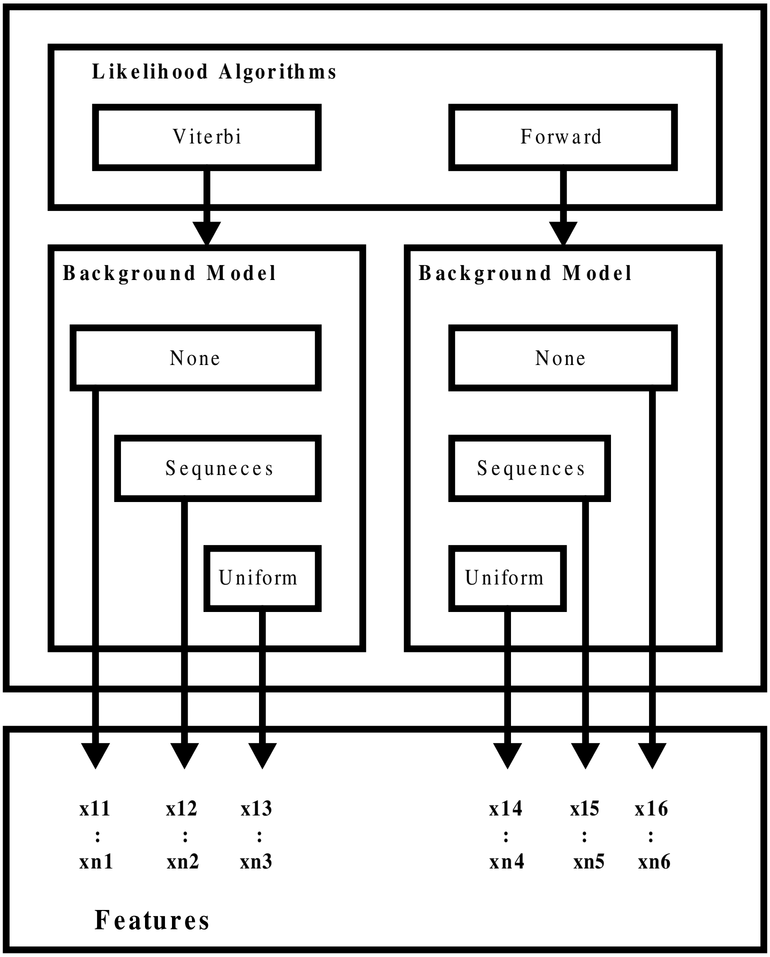

Fig. 1. Feature extraction process.

the training set. 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.[19] One of the significant parameters needed to tune the SVM is the choice of the *kernel function.* The kernel function allows SVM to locate the hyperplane in high dimensional space that effectively separate the training data. In our implementation, we use the Libsvm software implemented by Chih-Chung Chang and Chih-Jen Lin.[b] We primarily employed the Gaussian kernel for all classifiers. Radial basis functions have received significant attention, most commonly with a Gaussian of

[b] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

the form,

$$K(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \quad \text{for all } x, y \in X\,. \tag{6}$$

Classical techniques utilizing radial basis functions employ some method of determining a subset of centers.

We tested the performance of our technique on the SCOP database version 1.37 PDB90.[19] The use of SCOP (Version 1.37), datasets designed by Jaakkola *et al.*[11] allows direct comparison with the previous work on 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. Details of the datasets are available at http://www.cse.ucsc.edu/research/compbio/discriminative.

4. Results and Discussion

Figure 2 illustrates the overall performance of our protein remote homology detection method on the 33 test families. Figure 2 also shows the overall performance of other existing protein remote homology detection methods such as the SAM-T98 method as purely generative model and SVM-Fisher method as a discriminative model. Since most of the methods produce a probability score on a different scale, they cannot be easily compared. We report the 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 are testing. Values for the median RFP are shown on the X-axis, while on the Y-axis we plot the number of SCOP families, out of the 33 families that we tested. From Fig. 2, it is clear to notice that HMMcs achieved the lowest RFP values compared to all other methods.

From the previous literature we found that, the only results presented by Jaakkola and Haussler[10] to evaluate the performance of the SVM-Fisher were the Rate of False Positives (RFP), so the discriminate function is not used properly and hence the SVM margin is ignored, in this way, for instance, the discriminate function could classify every single data point as negative but the RFP could still be zero. By just using the RFP you are putting the margin right up against the negative data, therefore in this regard, we are presenting very favorable results in Figs. 3 and 4. The two figures illustrate two indices used to evaluate the accuracy — *sensitivity* and *specificity.* They describe how well a test discriminates between protein families. The sensitivity is the proportion of true positives, while the specificity is the proportion of true negatives.

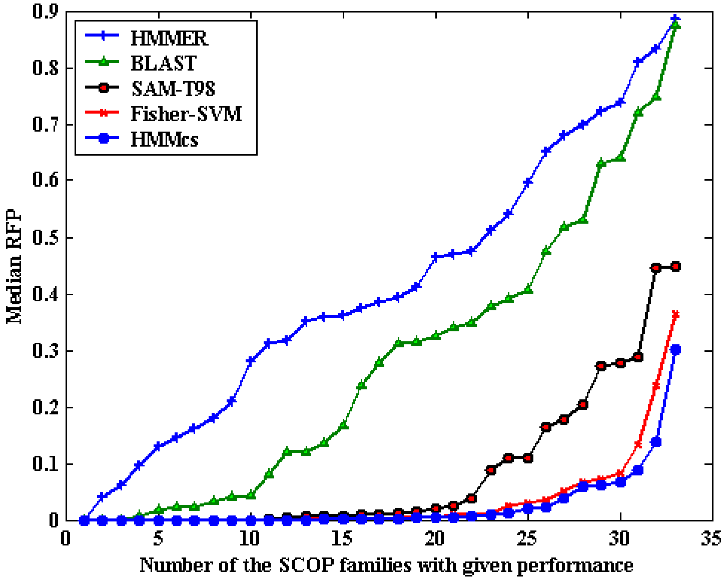

Fig. 2. Overall performance comparison.

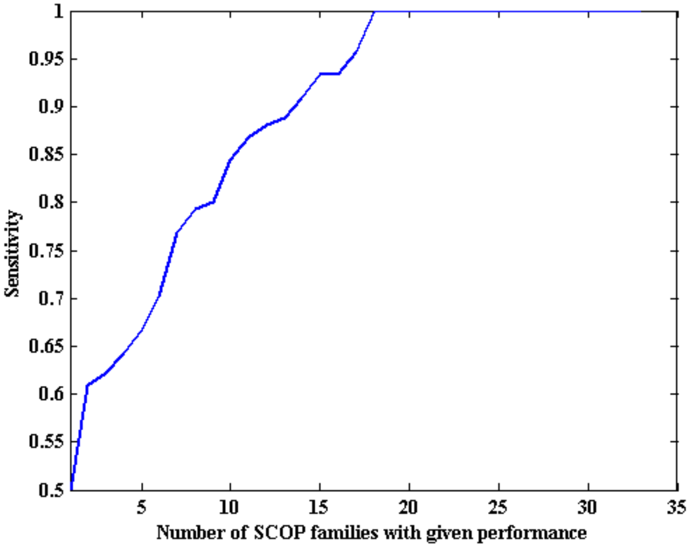

Fig. 3. Illustration of the sensitivity.

A Receiver Operating Characteristic (ROC) curve is also used to evaluate the HMMcs performance (as shown in Fig. 5). ROC is a graphical representation of the trade off between the false negative and false positive rates for every possible cut off. By tradition, the plot shows the sensitivity on the X-axis and the specificity on

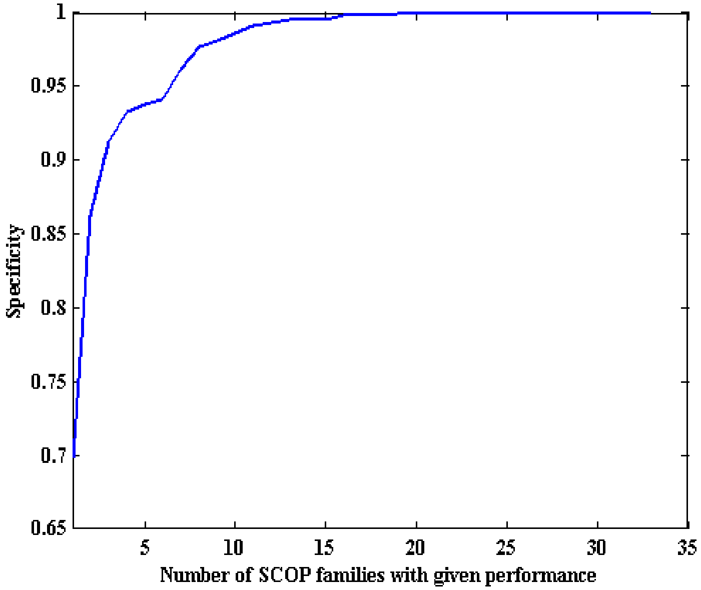

Fig. 4. Illustration of the specificity.

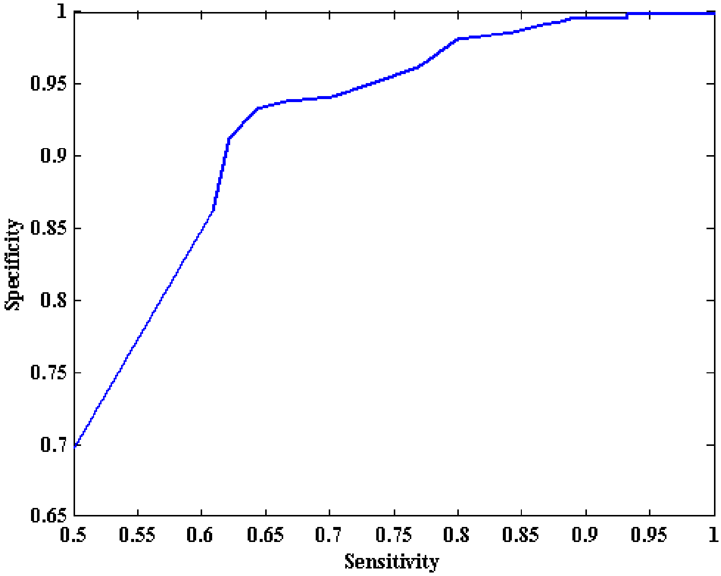

Fig. 5. ROC curve.

the Y-axis. The accuracy of a test is measured by the area under the ROC curve. An area of 1 represents a perfect test, while an area of 0.5 represents a worthless test. The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test; the true positive rate is high and the false positive rate is low. Statistically, more area under the curve means that it is identifying more true positives while minimizing the number/percent of false positives.

More detailed comparison is made between our approach and SVM-Fisher, method in terms of which method achieve lower RFP in all or most of the 33 families. Family-by-family comparison of the HMMcs performance against SVM-Fisher is shown in Fig. 6. Each family is plotted as a point (x, y) where x is the median RFP for the SVM-Fisher and y is the median RFP for the HMMcs method. From Fig. 6, it is clear to see the superiority of the HMMcs in achieving low RFP in most of the 33 families.

One significant characteristic of any protein remote homology detection algorithm is whether the method is computationally efficient or not. In this respect, the HMMcs algorithm is not significantly better than SVM-Fisher. Both algorithms include an SVM optimization, which is roughly $O(n^2)$, where n is the number of training set examples. The proposed method needs to perform 6 dynamic programming passes on each sequence for every two passes (forward and backward) that the Fisher kernel method performs. However, the computation of the kernel function in HMMcs is much faster and requires less memory since we deal with only six dimensions.

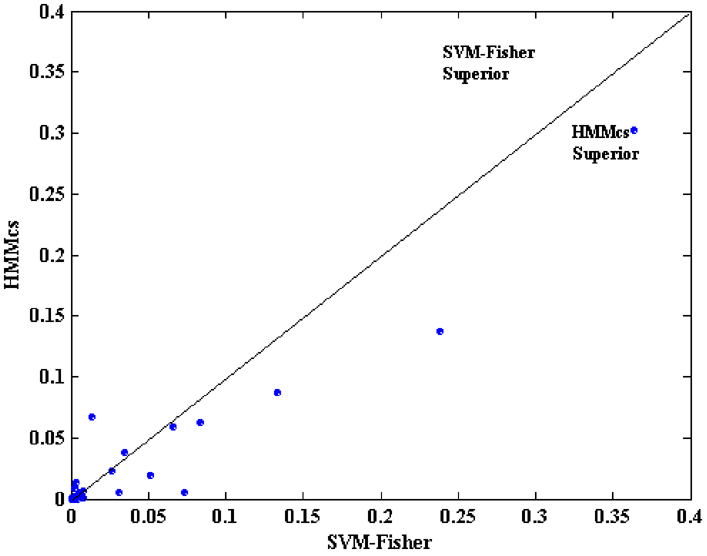

Fig. 6. Family-by-family comparison of the HMMcs against SVM-Fisher methods.

5. Conclusion and Future Work

In this paper, we have presented a new approach to recognize remote protein homologies. The main contribution of the approach is a simple method of constructing feature vectors and the combination of this representation with a classification method capable of learning in very sparse high-dimensional spaces. The method is a HMM based approach to combine different scoring algorithms. Combination of different scoring algorithms has proved efficient to increase the generalization performance of individual scoring methods. The experiments show that this method, which we call the HMMs combining score (HMMcs), improves on previous methods for classifying protein domains based on remote homology. Note that, recently, two more approaches for detecting remote protein homology have been developed and showed good performances.[20,21] The analysis and comparison of these methods as well as testing the HMMcs method over different datasets such as the one developed in Ref. 20 will be the subject of future research. Moreover, the proposed method has to be tested on other applications.

Acknowledgments

This work was supported in part by Malaysian Ministry of Science, under the Intensification of Research in Priority Areas (IRPA, Project No.: 74017). The authors thank Nello Cristianini (University of California, Davis) and Chih.-Jen Lin (National Taiwan University) for their valuable comments and cooperation.

References

1. C. Gibas and O. Jambeck, *Developing Bioinformatics Skills* (O'reilly, CA, 2001).
2. T. Smith and M. Waterman, Identification of common molecular subsequence, *J. Mol. Biol.* **147** (1981) 195–197.
3. W. R. Pearson, Rapid and sensitive sequence comparisons with FASTAP and FASTA, *Meth. Enzymology* **183** (1985) 63–98.
4. S. F. Altschul, W. Gish, W. Miller, E. Myer and J. Lipman, Basic local alignment search tool, *J. Mol. Biol.* **215** (1990) 403–410.
5. M. Gribskov, R. Lüthy and D. Eisenberg, Profile analysis, *Meth. Enzymol.* **183** (1990) 146–159.
6. A. Krogh, M. Brown, I. S. Mian, K. Sjölander and D. Haussler, Hidden Markov models in computational biology: Applications to protein modeling, *J. Mol. Biol.* **235** (1994) 1501–1531.
7. E. L. Sonnhammer, S. R. Eddy and R. Durbin, Pfam: A comprehensive database of protein domain families based on seed alignments, *Proteins* **28** (1997) 405–420.
8. S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller and D. J. Lipman, Gapped blast and Psi-blast: A new generation of protein database search programs, *Nucleic Acids Res.* **25** (1997) 3389–3402.
9. K. Karplus, C. Barrett and R. Hughey, Hidden Markov models for detecting remote protein homologies, *Bioinformatics* **14**(10) (1998) 846–856.

10. T. Jaakkola, M. Diekhans and D. Haussler, Using the Fisher kernel method to detect remote protein homologies, in *Proc. 7th Int. Conf. Intell. Syst. Mol. Biol.* Menlo Park, CA, 1999, pp. 149–158.
11. T. Jaakkola, M. Diekhans and D. Haussler, A discriminative framework for detecting remote protein homologies, *J. Comput. Biol.* **7** (1–2) (2000) 95–114.
12. V. N. Vapnik, *The Nature of Statistical Learning Theory* (Springer, New York, 1995).
13. N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines* (Cambridge University Press Cambridge, Cambridge, UK, 2000).
14. C. J. Merz and M. J. Pazzani, A principal components approach to combining regression estimates, special issue of *Mach. Learning*, **36** (1997) 9–32.
15. Y. Shimshoni and N. Intrator, Classifying seismic signals by integrating ensembles of neural networks, *IEEE-Signal Process.* **46** (1998) 1194–1201.
16. C. Merz, Using correspondence analysis to combine classifiers, *Mach. Learning* **36** (1999) 33–58.
17. S. L. Salzberg, D. B. Searls and S. Kasif, *Computational Methods in Molecular Biology* (Elsevier, 1998).
18. C. Barrett, R. Hughey and K. Karplus, Scoring hidden Markov models, *CABIOS* **13**(2) (1997) 191–199.
19. G. Murzin, S. E. Brenner, T. Hubbard and C. Chothia, SCOP: A structural classification of proteins database for the investigation of sequences and structures, *J. Mol. Biol.* **247** (1995) 536–540.
20. Li Liao and W. S. Noble, Combining pairwise sequence similarity and support vector machines for remote protein homology detection, *Proc. 6th Ann. Int. Conf. Res. Comput. Biol.* (*RECOMB'02*) Washington, DC, USA, 2002, pp. 225–232.
21. L. Christina, E. Eskin, J. Weston and W. S. Noble, Mismatch string kernels for SVM protein classification, December 2002, NIPS, Vancouver, pp. 9–14.