PROTEINS: Structure, Function, and Bioinformatics 57:518–530 (2004)

Remote Homolog Detection Using Local Sequence–Structure Correlations

Yuna Hou,[1*] Wynne Hsu,[1] Mong Li Lee,[1] and Christopher Bystroff[2]
[1]*School of Computing, National University of Singapore, Singapore*
[2]*Department of Biology, Rensselaer Polytechnic Institute, Troy, New York*

ABSTRACT **Remote homology detection refers to the detection of structural homology in proteins when there is little or no sequence similarity. In this article, we present a remote homolog detection method called SVM-HMMSTR that overcomes the reliance on detectable sequence similarity by transforming the sequences into strings of hidden Markov states that represent local folding motif patterns. These state strings are transformed into fixed-dimension feature vectors for input to a support vector machine. Two sets of features are defined: an order-independent feature set that captures the amino acid and local structure composition; and an order-dependent feature set that captures the sequential ordering of the local structures. Tests using the Structural Classification of Proteins (SCOP) 1.53 data set show that the SVM-HMMSTR gives a significant improvement over several current methods. Proteins 2004;57:518–530.** © 2004 Wiley-Liss, Inc.

Key words: remote homology; local structure; support vector machines; hidden Markov model; protein folding; I-sites; HMMSTR

INTRODUCTION

Breakthroughs in large-scale sequencing and the Human Genome Project have led to a surge in biological sequence information. Researchers are increasingly relying on computational techniques to cope with the massive amount of information generated. Homology detection is one such computational approach to interpret the protein sequences through the detection of homologous proteins.

Early methods in homology detection are based on pairwise comparisons of protein sequences using dynamic programming algorithms such as the Needleman–Wunsch[1] and Smith–Waterman[2] algorithms. Popular search tools such as BLAST[3] and FASTA[4] are fast approximations of these dynamic programming algorithms. However, these pairwise comparison methods do not work well for remote homologies.

In order to identify remote homologs, methods such as profiles for protein families,[5] hidden Markov models (HMMs),[6,7] and iterative methods such as PSI-BLAST[8] and SAM[9] have been introduced. The basic idea behind these methods is to generate a representative model for each protein family. Instead of comparing an unknown sequence to a specific protein sequence, we compare it to the generated model of the appropriate family. While these approaches provide better detection of the remote homologies as compared to the pairwise comparison methods, additional accuracy can be obtained by modeling the difference between positive and negative examples—positive if they are in the family and negative otherwise.

If a fixed number of descriptive parameters, or features, can be defined for each of the sequence families, then a general technique called a support vector machine (SVM) may be applied to find the boundaries between families in the space defined by the features. This method has been successfully applied to the homolog detection problem.[10–14]

An SVM algorithm chooses a hyperplane through the feature space that has a maximum margin between positive and negative examples. Each hyperplane divides one class of data (all of the sequences families of one fold type) from all of the other classes. A new data point (an unknown sequence) can then be classified by finding its relationship with each of the hyperplanes. SVM classifiers are better generalizers than neural networks for small data sets.[15]

The success of a SVM classification method depends on the choice of the feature set to describe each protein family. Methods that use only sequence information fail when the sequence similarity is very low, even if the two structures are very similar (see Fig. 1). In our earlier work, we noted that similar structures contain similar local structure motifs. Our first SVM method, SVM-I-sites,[16] demonstrated that the local structure content of a protein improves remote homolog detection, even without knowing the locations of the local features within the sequence. Local structure motifs were found using the I-sites library of sequence–structure correlations.[17] The I-sites library contains 262 short-sequence patterns that each has a strong correlation with the three-dimensional (3D) structure, locally. Our tests showed that SVM-I-sites was comparable in detection accuracy and more efficient than the state-of-the-art method SVM-pairwise.[11]

Further study revealed that I-sites motifs can occur in different orders along the sequence in different fold topologies. Thus, encoding structure features using only the

Correspondence to: Yuna Hou, School of Computing, National University of Singapore, Singapore 119260.
E-mail: houyuna@comp.nus.edu.sg

Received 18 January 2004; Accepted 28 April 2004

Published online 6 August 2004 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/prot.20221

© 2004 WILEY-LISS, INC.

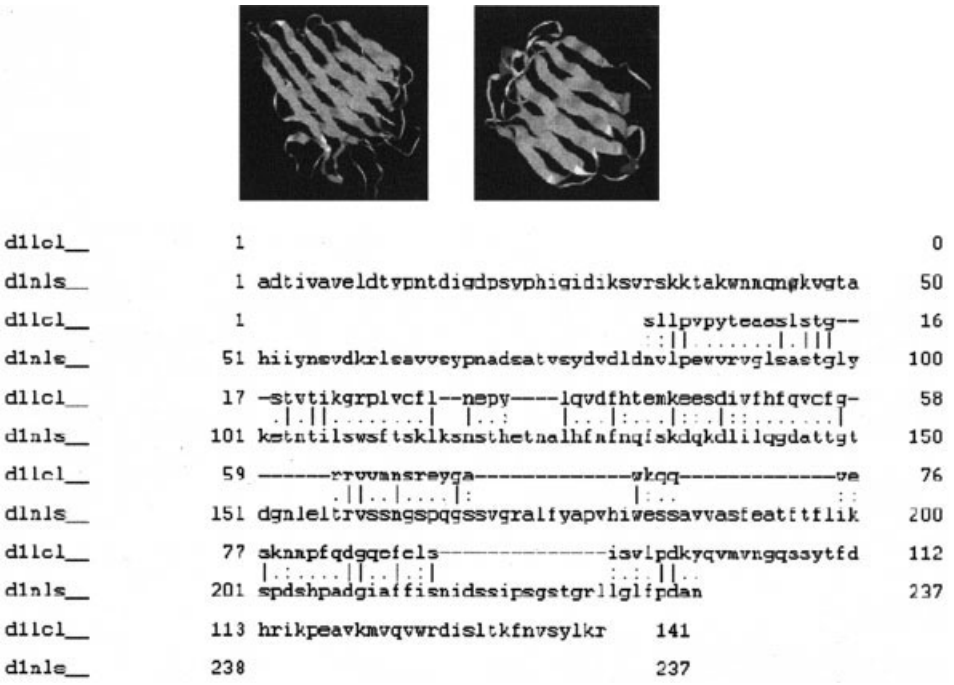

Fig. 1. Example of two proteins from SCOP superfamily 2.28.1. The structures are remarkably similar (top figures), but the alignment below shows only poor sequence similarity (10% identity).

composition of I-sites motifs may not uniquely define the global fold of the protein. The ordering of these motifs along the sequence also must play a role in dictating how the self-organization takes place during folding. Moreover, many of the I-sites motifs tend to overlap, resulting in redundant information.

Bystroff et al.[18] described a hidden Markov model called HMMSTR (Hidden Markov Model for protein STRucture) that extends and generalizes the I-sites library. HMMSTR models I-sites motifs as words of a higher order grammatical structure. Probabilistic transitions between I-sites motifs along the chain are thought to capture the propagation of structure during the folding process. A typical transition would be from helix to helix-cap, or β-strand to β-turn. Thus HMMSTR encodes some of the sequence ordering of local motifs. HMMSTR also removes the redundancy inherent in the I-sites model and reduces the number of free parameters. For example, several I-sites motifs are used to represent the different register shifts of the amphipathic heptad–repeat helix motif, while in HMMSTR these motifs are merged into a single, cyclic pathway of 7 Markov states.

For each query protein sequence, the output of the HMMSTR algorithm is a "γ" matrix,[19] where each element is the probability that a residue at a given position is associated with one of the Markov states. The states are conserved in remote homologs, because the local structure elements are conserved, and the relative positions of the states along the sequence are approximately conserved, because remote homologs conserve topology. Both the state and position information are crucial for defining the similarity of two proteins.

In this article, we investigate ways that the position-specific local structure information in HMMSTR can be efficiently encoded as fixed-length feature vectors for an SVM. There are two issues to address here:

1. The length of γ matrix is variable, while the input to the SVM must be a fixed-length vector. We encode the sequence order of the local motifs by performing alignments with the database, using a novel HMMSTR-based similarity score. Each alignment score is one feature.
2. The γ matrices are high dimensional. This incurs high computational costs when encoding the state and position information. We overcome this problem by first performing a dimension reduction before attempting to align the γ matrices of two proteins.

These feature vectors are subsequently used to train an SVM, SVM-HMMSTR. In short, for each protein, we obtain 2 sets of features. The first set aims to capture the amino acid and local structure composition, while the second set encodes the alignment scores. The new SVM classifier was tested on the Structural Classification of Proteins (SCOP) 1.53 data set and significantly outperformed existing methods.

MATERIALS AND METHODS

HMMSTR Model

HMMSTR is a better model for local structure prediction than the I-sites library, improving 8-residue fragment prediction accuracy from 43% to 59%.[18] This is because HMMSTR models the possible ways of arranging sequence–

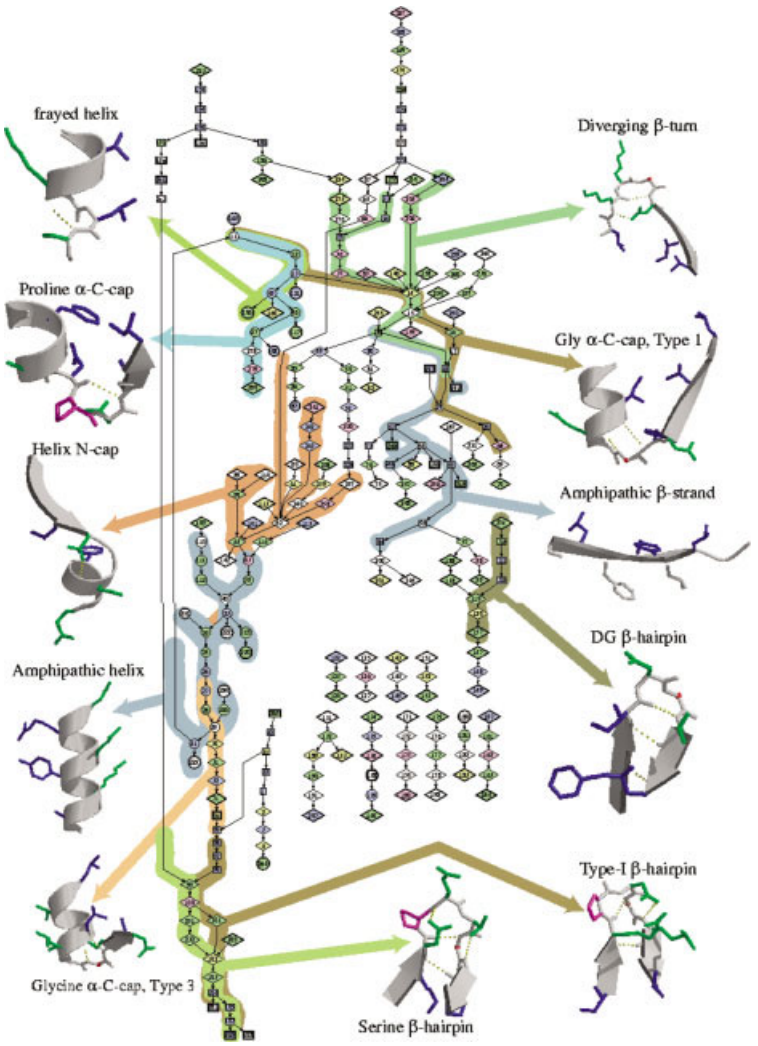

Fig. 2. HMMSTR model built from I-sites Library. Symbols represent hidden states: circles, predominantly helix; squares, strand; diamonds, loop or turn; yellow, glycine; magenta, proline; blue, nonpolar; green, polar; white, no predominant amino acid. Only high probability connections are shown (Reprinted by permission of the authors[18]).

structure motifs along the sequence, and because overlapping motifs have been merged, reducing redundancy and increasing the statistical significance of the amino acid profiles.

The topology of HMMSTR is a highly branched and multicyclic network. Each of the 262 I-sites motifs is represented as a chain of Markov states, which contains information about the sequence and structure attributes of a single position in the motif. Adjacent positions are modeled by directionally linked states. A hierarchical merging of these chains of states, based on sequence and structure overlaps, resulted in a graph that contains almost all the motifs. The merged graph of I-sites motifs comprises a network of states connected by probabilistic transitions, or a HMM, as shown in Figure 2.

In contrast to the more familiar family-specific HMMs,[20] HMMSTR captures simultaneously the recurrent local features of sequences and structures across all families of globular proteins. Each state in HMMSTR can produce, or "emit," amino acids and structure symbols according to a probability distribution specific to that state. There are 4 probability distributions defined for the states in HMMSTR—b, d, r, and c—that describe the probability of observing a particular amino acid, secondary structure, backbone angle region, or structural context descriptor, respectively. This set of emission probabilities for a given state q_i is collectively called B_{q_i}. The values $b_{q_i}(\psi)$ $(1 \leq \psi \leq 20)$ are associated with probabilities for the emission of amino acids. The values $d_{q_i}(\psi)$ $(1 \leq \psi \leq 3)$ are the probabilities of emitting helix, strand, or loop, respectively. The values $r_{q_i}(\psi)$ $(1 \leq \psi \leq 11)$ are the probabilities of emitting one of the 11 dihedral angle symbols. Finally, $c_{q_i}(\psi)$ $(1 \leq \psi \leq 10)$ are probabilities of emitting 1 of 10 structural context symbols.

The database used for training, evaluation, and testing of the HMMSTR is encoded as a linear sequence of amino acids and structural observables. The amino acid sequence data consist of a parent ("parent" means the sequence

upon which the multiple alignment is based) amino acid sequence of known 3D structure and an amino acid profile obtained by alignment to the parent sequence. The amino acid of the parent sequence is denoted by O_t, and the profile by $\{O_t^\psi\}$ $(1 \le \psi \le 20)$. For each position t, there are 3 structural identifiers: 3-state secondary structure D_t, discrete backbone angle region R_t, and the context symbol C_t (context symbols C_t were assigned to strands and loops in the training sequences based on the nonlocal context of position t; for example, loops were "hairpins," "diverging turns," "corners," one of two types of helix cap, or just "coil"; also, strands could be "middle" of the sheet or "end"). Therefore, any sequence s of length T is given by the values of the attributes at all positions $s_t = \{O_t, \{O_t^\psi\}, D_t, R_t, C_t\}$ $(1 \le t \le T)$.

HMMSTR models database sequences based on the notion of a path. A path is a sequence of states through the HMMs, denoted $Q = q_1 q_2 \ldots q_T$. The probability of a sequence s given the model λ, $P(s \mid \lambda)$, is obtained by summing the relevant contributions from all possible paths Q:

$$P(s \mid \lambda) = \sum_{allQ} \pi_{q_1} B_{q_1}(s_1) a_{q_1 q_2} B_{q_2}(s_2) \cdots a_{q_{T-1} q_T} B_{q_T}(s_T), \quad (1)$$

where a_{qiqj} $(1 \le i, j \le T)$ represents the probability of a transition from state q_i to state q_j, π_{q_i} is the probability of initiating a sequence at state q_1 and $B_{q_i}(s_t)$ is the probability of observing s_t at state q_i, which for observation of a single sequence is given by

$$B_{q_i}(s_t) = \begin{pmatrix} d_{q_i}(D_t) \\ r_{q_i}(R_t) \\ c_{q_i}(C_t) \end{pmatrix} b_{qi}(O_t). \quad (2)$$

Usually, only one of the structural emission symbols d, r, or c is included in B_{q_i} in any given training run. However, in principle, any combination could be used. For remote homology detection, we used a model trained on r for the prediction, because this is the most closely tied to the I-sites library representation. We found significant improvements in performance when we used amino acid profiles instead of single amino acid sequences for training and for subsequent predictions. For the probability of observing a given profile O_t^ψ at position t in a sequence, we use the multinomial distribution, and the expression for B becomes

$$B_{q_i}(s_t) = r_{q_i}(R_t) \sum_{\psi=1}^{20} b_{q_i}(\psi)^{Ncount \times O_\psi^t}. \quad (3)$$

In this equation, *Ncount* is the "depth" of the multiple sequence alignment, which is roughly the number of homologous sequences in the alignment. However, this number is an artifact of uneven sampling of sequence families in the database. To give equal weight to all sequence families in the training set, *Ncount* was taken to be a fixed value.

To use the HMMSTR model, we input a single sequence to predict its Markov states, as follows: We run PSI-BLAST against the Swissprot database to generate a multiple sequence alignment. Then, the multiple sequence alignment is converted to a sequence profile. Next, the profile is aligned to HMMSTR to get a probability distribution over all states at each position, that is, the γ matrix. This matrix is computed by the Forward–Backward Algorithm[19] and describes the probability of each HMMSTR state at each position; that is,

$$\gamma_{pq} = P(q \mid s, t, \lambda) \quad (4)$$

for all the 281 HMMSTR states $(1 \le q \le 281)$ and for all residues s_t $(1 \le t \le T$, where T is the length of the protein). The γ matrix may be viewed as the translation of a given sequence to a language of I-sites motif descriptors. HMMSTR is a grammatical model for that motif "language."

Feature Extraction and Representation

There are two schools of thought in defining the similarity of proteins.[21] The local view considers only 10–20% of residues to be critical, while the global view advocates that similarities should occur along the entire sequence. In order to accommodate both views, we define two sets of features: One feature set captures the composition of a protein in terms of the local structure motifs and amino acids regardless of their sequential ordering, and the other feature set only considers the longest conserved regions between two proteins by encoding both the composition and the arrangement order of the regions. These two feature sets are called "order-independent" and "order-dependent," respectively. This section presents the details of our feature extraction and representation procedure. The objective is to capture both the state information and the positional information in terms of fixed length vectors to be used for training an SVM.

Order-independent feature set

The order-independent feature set aims to capture the composition of amino acids and local structure. Each feature is indexed by (x,s), where x represents the 20 amino acids and s is the 281 Markov states in HMMSTR. $\xi(x,s)$ is the total number of x in state s summed over all positions of a protein. Therefore, there are a total 20 * 281 features. The feature value $\xi(x,s)$ can be computed from the gamma matrix:

$$\xi(x, s) = \sum_{\substack{t=1 \\ O_t = x}}^{T} \gamma_t(s), \quad (5)$$

which is the sum of all the probabilities of being in state s and observing amino acid x across all the positions of the target protein.

The feature set given by Eq. (5) is a variation of the Fisher score defined by Jaakkola et al.[10] Fisher score vector U_x can be obtained by the following formula:

$$U_x = \nabla_{\theta_{x,s}} \log P(X \mid H_1) \quad (6)$$

Each component of U_x is a derivative of the log-likelihood score for the query sequence X with respect to a particular emission probability parameter $\theta_{x,s}$ of the HMM H_1. The magnitude of the components of U_x specifies the extent to which each parameter $\theta_{x,s}$ contributes to generating the query sequence. The derivatives of $\log P(X \mid H_1)$ with respect to the $\theta_{x,s}$ as the components of the Fisher score vector U_x can be computed by the following formula[10]:

$$\frac{\partial}{\partial_{\theta_{x,s}}} \log P(X \mid H_1) = \frac{\xi(x, s)}{\theta_{x,s}} - \xi(s), \quad (7)$$

where $\theta_{x,s}$ is one emission probability parameter that specifies the probability of emitting amino acid x in state s, and $\xi(s)$ is the expected number of times we visit state s as we traverse all paths through the model.

In this work, we use a sequence profile instead of a single amino acid sequence. Therefore, for the order-independent feature set, we define a variation of the Fisher score to capture the expected sum, over all the positions of a protein, of the position-specific frequency of visiting state s and emitting symbol x.

Order-dependent feature set

The relative position of local structure is lost in the order-independent feature set. Therefore, a strong similarity between two such feature vectors tells us only that the two sequences have the same local structure motifs. It does not tell us that the motifs are arranged in the same order along the sequence. For example, two proteins that are predominantly helical would have similar "local" features, even if they are globally different. Although to some extent, the existence of conserved helix-caps or other, less common motifs makes this feature space more selective than amino acid or secondary structure composition measures.

In this section, we define an order-dependent feature set by aligning the γ matrices to include both the states' posterior probability and their positions in the protein. The order-dependent feature set for a protein X is defined as $F_x = (f_{x_1} f_{x_2}, \ldots, f_{x_\omega})$, where ω is the total number of proteins in the training set and f_{x_i} is the E-value of the Smith–Waterman score between protein X and the ith training set protein. This feature set has a similar format as the features in SVM-pairwise method. The difference is that, in this work, f_{x_i} is the E-value of the alignment score of 2 γ gamma matrices, rather than the E-value of the Smith–Waterman score between 2 sequences in SVM-pairwise method.

In the following sections, the procedure of deriving the E-values of the alignment of 2 γ matrices is described. This process comprises 3 steps, namely, dimension reduction, similarity computation of 2 γ matrices, and transformation of similarities scores into E-values.

Dimension reduction. Recall that the γ matrix is 281 rows by T columns, where T is the length of the protein sequence. We observe that only a few of the 281 states have probability values that are significantly greater than zero in most of the columns in the γ matrices. Most of the states have probability values that are very close to zero. A state probability value is defined as "significant" if it occurs by chance with a probability less than or equal to a predefined probability value α. Given a significance value α, we use the following steps to determine the cutoff value and the maximum number of states C needed to store the significant states for each column of a γ matrix. This procedure is based on a sample set of γ matrices for proteins in the SCOP database[22] version 1.53:

1. Sort all the states in a descending order by their probabilities. The cutoff is the probability value of the nth state, where $n = N * \alpha$, and N is the number of the sampled states.
2. Count the number of states in each column with a probability value greater than this cutoff, and C is the maximum of these numbers.

TABLE I. Correlation Between *C* and α

Probability value α	Cutoff	Number of States *C*
0.001	0.71	1
0.005	0.16	4
0.01	0.06	8

Table I lists the cutoff values corresponding to the different probability values. The variable C represents the maximum number of states in one column in these samples with values greater than the cutoff. It is clear that for an α of 0.01, it is sufficient to just maintain the top 8 states.

This motivates us to reduce the dimensionality of the γ matrix by storing only the top C states for each column, and the remaining states are assumed to have probability 0. In our experiments, we set C at 10 to ensure that all the significant values for $\alpha \leq 0.01$ are included. With this, we transform each γ matrix into a record with the format: "T ($S1$ $V1$ … SC VC)," where T is the length of the γ matrix, the set ($S1$ $V1$ … SC VC) denotes the concatenation of C entries of each column in the γ matrix, S_i and V_i ($1 \leq i \leq C$) represents the state number from 1 to 281, and the probability value corresponds to the state. Subsequent similarity comparisons are carried out on this transformed data set.

Similarity computation. We align the extracted γ matrices using Smith–Waterman algorithm to get a similarity score to measure the overall conserved folding similarity of two proteins. In order to use the Smith–Waterman algorithm for alignment, a set of parameters (similarity score, gap penalties) called a scoring scheme must be defined. Existing scoring schemes such as BLOSUM or PAM are designed for sequence alignment, and are not applicable for the γ matrices alignment problem. Hence, we redefine the scoring scheme for γ matrices comparison. Yona and Levitt[23] examined the principles for defining a new scoring scheme for the Smith–Waterman algorithm for a new alignment problem. They concluded that in order to be consistent with the BLOSUM and PAM scoring schemes, the similarity score of two positions (a,b) must satisfy the conditions:

1. $mean\{Score(a,b)\} < 0$
2. $max\{Score(a,b)\} > 0$.

The first condition guarantees that the average score of a random match is negative, while from the second condition, a match with a positive score is possible. Based on these two principles, we define the scoring scheme for our γ matrix alignment problem. The following subsection presents the details of the procedure for redefining the new similarity score of two positions.

Similarity score. To encode the position information, we need to compute the similarity of 2 γ matrices. This involves performing a Smith–Waterman alignment on the 2 γ matrices. To do this, we must first reduce the 2 matrices to a pairwise similarity matrix. In other words, one pair of columns is first reduced to a single similarity value.

Given 2 γ matrix columns p and q, where the top C states are stored, the similarity score of two columns is defined as the cosine similarity score of the two columns. The main reason we use a cosine similarity score in place of a symmetrized relative entropy, which is used by Yona and Levitt,[23] is that multiplication is a much faster operation compared to the log operation; especially the time needed for such operations is huge. The cosine similarity score is given by the formula

$$(p, q) = \frac{p \cdot q}{\sqrt{(p \cdot p)(q \cdot q)}} \tag{8}$$

where " $\cdot$ " denotes "dot product," and the dot products (i.e., $p \cdot q$, $p \cdot p$, and $q \cdot q$) in Eq. (8) are all computed with the stored top C states under the assumption that the remaining states are insignificant. The computation of the self-dot products $p \cdot p$ and $q \cdot q$ is very straightforward, which is again over just the top C states. The dot product of $p \cdot q$ is equal to the sum of the products over just the states that overlap between p and q.

Once we have computed the similarity scores of all the pairwise columns of the 2 γ matrices, the next step is to map the computed similarity scores into a new range that guarantees the average score of a random match is negative, and at the same time, a match with a positive score is possible. A simple method that can map the similarity score defined in Eq. (8) into a new range that satisfies the two conditions is to subtract the similarity scores with a shift value. To determine the shift value, we first compute the average similarity score calculated for a large set of γ matrix column pairs, which is 0.05. To ensure that the average score of a random match is negative, the shifted value much be larger than 0.05. To ensure that a match with a positive score is possible, the maximum shifted value is less than 1. Shift values ranging from 0.1 to 0.9 are investigated in our experiments.

Gap penalties. In addition to the shift value, gap penalties also play an important role in deriving a sensitive scoring scheme. In order to determine the optimal shift value and gap penalty, a total of 9 £ 3 sets of parameters are tested, with shift value between 0.1 and 0.9 (values of 0.1, 0.2, 0.3, …, 0.9), a gap opening penalty between 1 and 3 (values of 1, 2, 3), and gap extension penalty between 0.1 and 0.3 (values of 0.1, 0.2, 0.3). In keeping with the BLOSUM62 matrix, we set the gap extension penalties to be one order of magnitude smaller than gap open penalty.

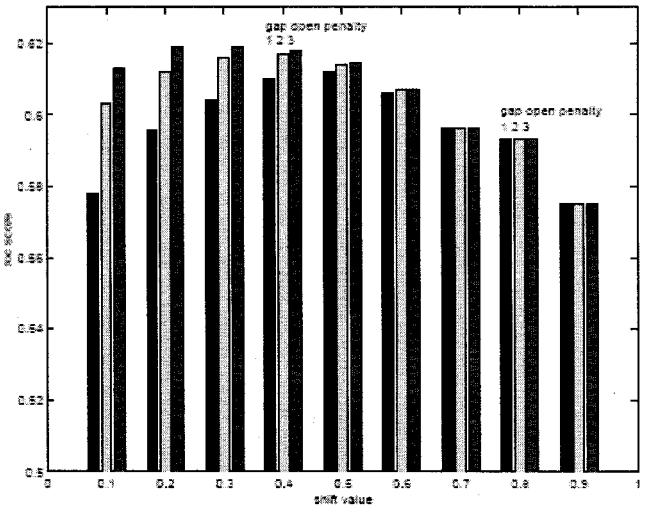

Fig. 3. Performance with different parameter sets. The *x* axis is the shift value; the *y* axis is the ROC score computed for the sampled protein pairs. For each shift value, there are three ROC scores that correspond to the gap opening penalty 1,2,3 and gap extension penalty 0.1,0.2,0.3, respectively.

We sample a subset of protein pairs that are either related or unrelated from SCOP database[22] version 1.53 to determine the best scoring scheme. We refer to two proteins within the same superfamily in SCOP as "related," and otherwise, as "unrelated." Given a specific set of parameters, we first run Smith–Waterman algorithm using these parameters as the scoring scheme to obtain the similarity score for all the sampled protein pairs. The receiver operating characteristic (ROC) score[24] is computed with all the alignment scores of the protein pairs and their relatedness (1 for related and 0 for unrelated) as the input to measure the goodness of this set of parameters. Figure 3 shows the results of the experiments. From the graph, we set the shift value at 0.3, with the gap opening penalty of 3 and gap extension penalty of 0.3.

After redefining the scoring scheme, we run the standard Smith–Waterman algorithm to get the similarity score of the 2 γ matrices.

Transformation of similarity score into *E*-value. Empirical studies[25] have shown that the distribution of local gapped similarity scores can be well approximated by the extreme value distribution.[26] We transform the scores into a statistical significance value called "*E*-value" to distinguish true similarities from random matches. The *E*-value can be computed using the formula $E = Kmne^{-\lambda S}$, where S denotes the similarity score, m and n are the lengths of the compared proteins, and λ and K are two empirically derived parameters.

We use the direct estimation method[25] to estimate the parameters λ and K. We collect the scores of 5000 optimal alignments of γ matrices. These alignments are produced from random protein sequences of length $n = m = 900$

using the scoring scheme described in the previous subsection. The number of alignments that score above a given threshold τ are counted. It is suggested in Waterman and Vingron[25] that the probability of an alignment scoring less than or equal to τ is given by $\exp(-\lambda mnK^{\tau})$. After an appropriate transformation $[log\{-log(data)\}]$, the empirical distribution function is expected to form a straight line, which facilitates the estimation of the parameters λ and K. However, this estimation does not take into consideration the length of the real data that will have an effect on the value of parameter λ.[25] To eliminate this effect, λ is corrected by using the estimated λ and K to search the protein database using the maximum likelihood estimation (MLE) method.

Data Sets

We assess the recognition performance of each algorithm by testing its ability to classify protein domains into superfamilies in the SCOP[22] version 1.53. The same experiment setup as the SVM-pairwise[11] method is adopted here to allow for a direct comparison. Remote homology is simulated by holding out all members of a target SCOP family from a given superfamily as follows:

1. Close sequences are removed using an E-value threshold of 10^{-25}, and this resulted in 4352 distinct sequences, grouped into families and superfamilies.
2. Positive training examples are chosen from the remaining families in the same superfamily, and negative test and training examples are chosen from outside the target family's fold. The held out family members serve as positive test examples. A total of 54 families containing at least 5 family members (positive test) and 10 superfamily members outside of the family (positive train) are produced. For each family, negative examples are taken from outside of the positive sequences' fold, and are randomly split into training and testing sets in the same ratio as the positive examples.

Metrics

To assess the performance of a remote homology detection method, we consider two metrics: the Receiver Operating Characteristic (ROC50) score[24] and median Rate of False Positives (RFP).[10] The ROC score combines measures of a search's sensitivity and specificity. The ROC score is the area under a curve that plots true positives versus true negatives for varying score thresholds, and it measures the probability of correct classification. Therefore, a score of 1 indicates perfect separation of positives from negatives, whereas a score of 0 denotes that none of the sequences selected by the algorithm is positive. The ROC50 score is the area under the ROC curve, up to the first 50 false positives. ROC50 has the advantage of providing a more efficient and sensitive way to evaluate different methods over the ROC score. The median RFP score is the fraction of negative test sequences that score as high or better than the median-scoring positive test sequence. Median RFP is used to measure the error rate of the prediction under the score threshold where half of the true positives can be detected. These measures were used for evaluation in Jaakkola et al.[10] and Liao and Noble.[11]

Support Vector Machines

SVMs[15,27] have strong theoretical foundations and excellent empirical successes. The SVM is a supervised machine learning method that developed rapidly and has been widely used in many kinds of pattern recognition problems. The basic method of SVM is to transform the samples into a high-dimension Hilbert space and to seek a separating hyperplane in this space. The separating hyperplane, which is called the optimal separating hyperplane, is chosen in such a way as to maximize its distance from the closest training samples. The SVM usually outperforms other machine learning technologies, including Neural Networks and K-Nearest Neighbor classifiers.[15]

We use the Gist publicly available SVM software implementation (http://microarray.cpmc.columbia.edu/gist/index.html), which implements the soft margin optimization algorithm described in Jaakkola et al.[10] The base kernel is normalized so that each vector has length 1 in the feature space, that is, $K(X,Y)=X \cdot Y/\sqrt{(X \cdot X)(Y \cdot Y)}$. This kernel $K(.,.)$ is then transformed into a radial basis kernel.

RESULTS AND DISCUSSION

Setup of Competing Methods

We compare SVM-HMMSTR with 5 methods: the generative models PSI-BLAST and SAM, and the SVM-based discriminative models SVM-I-sites, SVM-pairwise, and SVM-Fisher. We first briefly describe the setup procedure of these methods.

It is not straightforward to compare PSI-BLAST and SAM, which requires as input a single sequence, with methods such as SVM-Fisher and SVM-HMMSTR, which take multiple input sequences. We address this problem by randomly selecting a positive training set sequence to serve as the initial query.

PSI-BLAST is run for two iterations on the Swissprot database with an E-value threshold 0.001, and the resulting profile is then used to search against the test set sequence. The resulting E-values are used to rank the test set sequences. The PSI-BLAST settings used here are exactly the same as the ones we used to build the profiles for our SVM-HMMSTR method.

For the SAM method, HMMs were trained using the Sequence Alignment and Models toolkit (www.soe.ucsc.edu/research/compbio/sam.html).[7] The generative models were trained from an existing library of SAM-T99 HMMs. The SAM-T99 algorithm, described more fully in Karplus et al.,[9] builds an HMM for a SCOP domain sequence by searching the nonredundant protein database Swissprot for a set of potential homologs of the sequence and then iteratively selecting positive training sequences from among these potential homologs and refining a model. The resulting model is stored as an alignment of the domain sequence and final set of homologs. Once a model is obtained, it is straightforward to compare the test sequences to the model and the resulting reverse scores are used to rank the test set sequences.

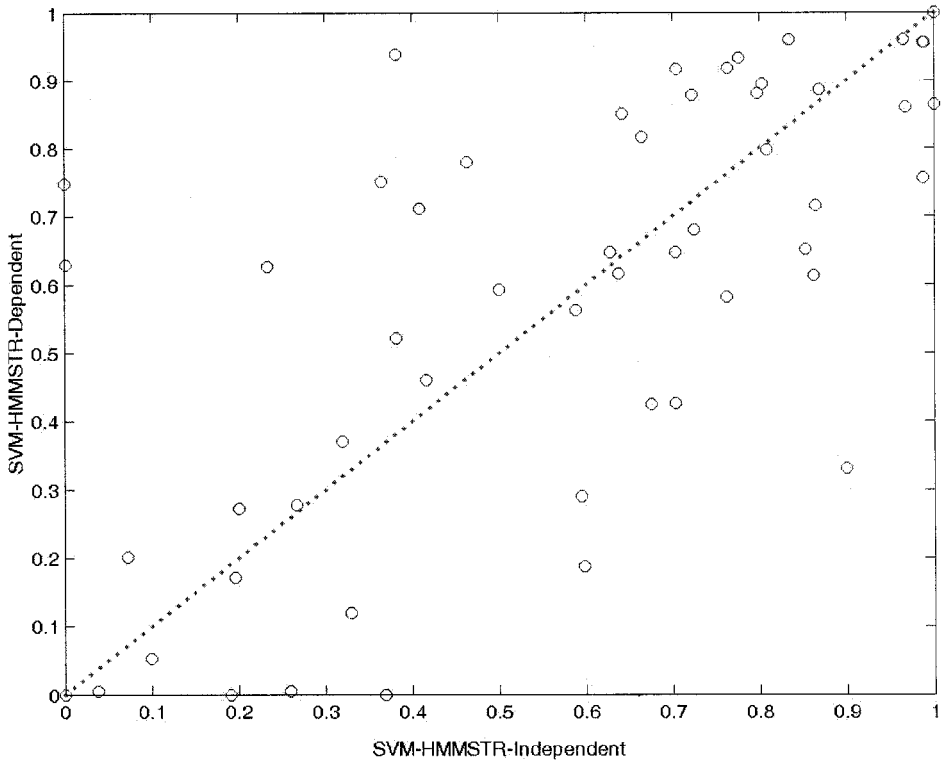

Fig. 4. Family-by-family comparison of SVM-HMMSTR-Independent and SVM-HMMSTR-Dependent. The coordinates of each point in the plot are the ROC50 scores for one SCOP family, obtained using SVM-HMMSTR-Independent and SVM-HMMSTR-Dependent. The dotted line is $y = x$.

For the SVM based methods, SVM-Fisher, SVM-pairwise, and SVM-I-sites, the key steps are feature representation and extraction. The SVM-Fisher method uses feature vectors calculated from the parameters of a profile HMM. For each positive training set, a HMM is first built with the SAM package as described above, and subsequently used to compute a vector representation for any sequence using the accompanying program *get_fisher_scores* in the SAM package. SVM-pairwise uses the pairwise Smith–Waterman sequence similarity algorithm in place of the gradient vector in the SVM-Fisher method that we described earlier. In contrast, SVM-I-sites encodes the local structure composition of a protein as the sum of I-sites motif confidence scores,[16,17] where each motif defines one feature. After the vectorization step, all of the SVM-based methods will define a similarity score for 2 proteins based on the feature vectors and use that similarity as the kernel of the classifier.

Results

We first investigated the effect of varying the composition of the feature sets on the detection accuracy. We call the method that uses only the order-independent feature set as "SVM-HMMSTR-Independent," and the method using only the order-dependent feature set as "SVM-HMMSTR-Dependent." Figure 4 shows a family-by-family comparison of the performances of the two methods.

From Figure 4, we realize that "SVM-HMMSTR-Independent" and "SVM-HMMSTR-Dependent" are complementary. This suggests that the local structure composition and the sequential order of the local motifs are equally important in determining the similarity of 2 proteins and a combination of the 2 feature sets should achieve a better performance than either feature set alone.

There are 2 basic approaches for combining the 2 feature sets. The most direct method is to concatenate the 2 feature sets into a longer feature set. We call this method "SVM-HMMSTR-Hybrid." Another approach is to first obtain 2 classifiers with the 2 feature sets individually and then combine their results, either taking the average or maximum for each query protein. We refer to these methods as "SVM-HMMSTR-Ave" and "SVM-HMMSTR-Max," respectively.

Table II summarizes the performance of the various methods we tested in terms of ROC50 and median RFP scores averaged over all 54 families tested. Since SVM-HMMSTR-Ave gave the best performance, we will use SVM-HMMSTR-Ave for the rest of the comparison tests.

The distribution of ROC50 and median RFP scores are shown in Figures 5, 6, and 7. In each case, a higher curve corresponds to more accurate remote homology detection performance. Using either performance measure, the SVM-HMMSTR method performs significantly better than the other 5 methods.

We also assess the statistical significance of differences among methods using Wilcoxon signed rank test.[13] The resulting p values are adjusted using a Bonferroni correction for multiple comparisons. As shown in Table III,

SVM-HMMSTR significantly outperforms all of the other methods with a p value 0.05.

Many of these results agree with previous assessments. For example, the relative performance of SVM-Fisher and SAM agrees with the results given in Jaakkola et al.,[10] as does the relative performance of SAM and PSI-BLAST with the results given in Park et al.[28] and the relative performance of SVM-I-sites and SVM-pairwise given in Hou et al.[16]

One surprise is the magnitude of the difference between SVM-pairwise and the 2 methods, SVM-Fisher and SAM, that directly or indirectly utilize SAM-T99 model. It is reported in Liao and Noble[11] that SVM-pairwise significantly outperforms these 2 methods, which is not the case in this work. This difference can be explained as follows: Our experiments allowed the iterative algorithm to access to all of Swissprot database during training of the model, while in Liao and Noble,[11] they are only given a small training set (containing only a handful of positive samples), without the benefits of the potential homologs in the Swissprot database.

TABLE II. ROC50 and Median RFP Averaged Over 54 Families for Different Methods

Methods	Mean ROC50	Mean median RFP
SVM-HMMSTR-Ave	0.640	0.038
SVM-HMMSTR-Max	0.618	0.043
SVM-HMMSTR-Hybrid	0.617	0.048
SVM-HMMSTR-Independent	0.572	0.051
SVM-HMMSTR-Dependent	0.587	0.048
SVM-I-sites	0.466	0.073
SVM-pairwise	0.438	0.094
SVM-Fisher	0.437	0.123
SAM	0.374	0.230
PSI-BLAST	0.264	0.336

The most significant result from our experiments is the top-ranking performance of the SVM-HMMSTR method. This result is further illustrated in Figure 8, which shows a family-by-family comparison of the 54 ROC50 scores computed for SVM-HMMSTR and SVM-pairwise method.

Our order-independent feature set uses a variation of the Fisher score, and it plays essentially the same role as the Fisher score features in that both capture the similarity of 2 proteins in terms of the composition of HMM states and amino acids. At the same time, the order-dependent feature set is defined by the E-value of an alignment score between 2 HMMSTR-derived profiles, which differentiates our order-dependent feature set from that of the SVM-pairwise algorithm. SVM-pairwise uses the E-value of a Smith–Waterman alignment score. Thus, we can surmise from the results shown in Table II that local structure–based HMMSTR information provides better features than the sequence-based profile HMMs, and alignment of HMMSTR-derived profiles provide better features than alignment of sequences. Figure 9 is the ROC50 distribu-

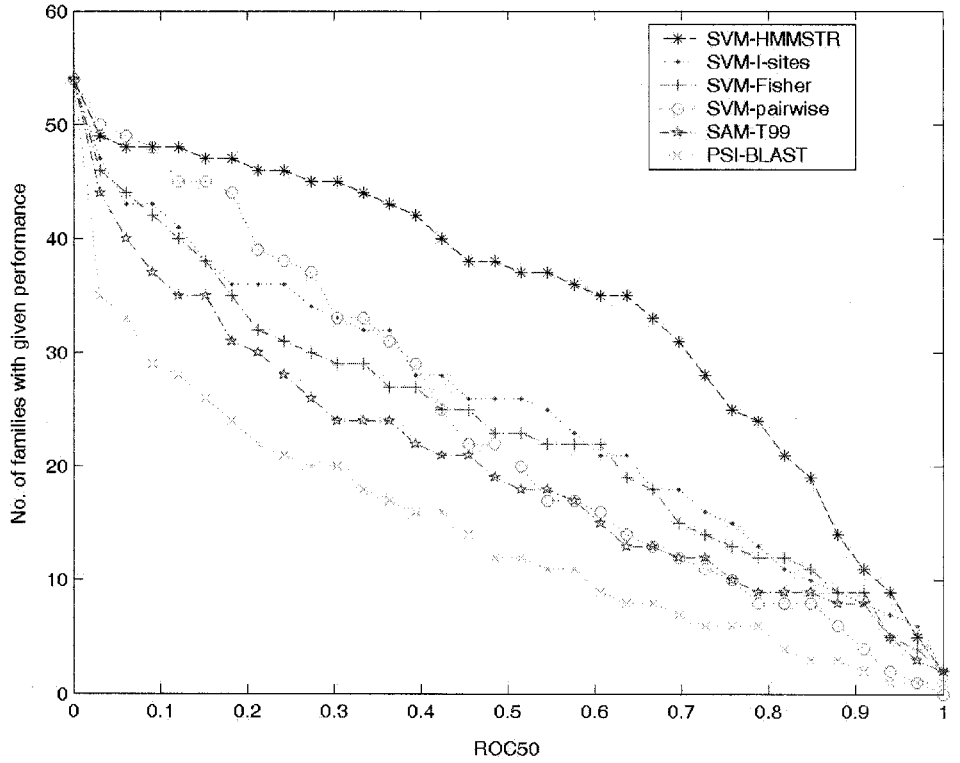

Fig. 5. Relative performance of homology detection methods. The graph plots the total number of families for which a given method exceeds a ROC50 score threshold. Each series corresponds to one of the homology detection methods described in the text.

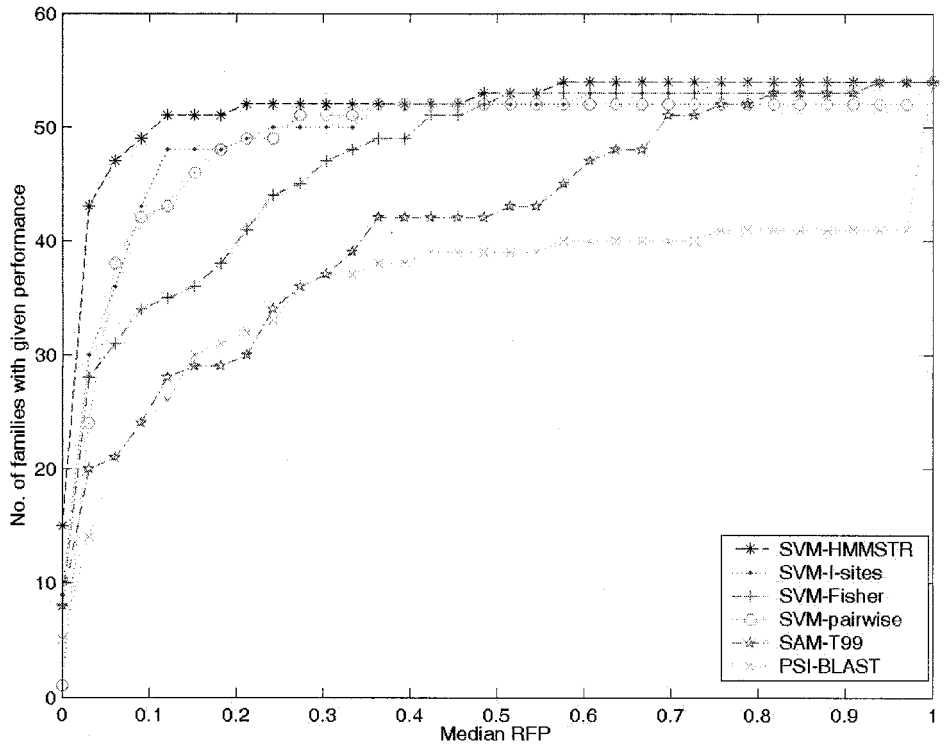

Fig. 6. Relative performance of homology detection methods. The graph plots the total number of families for which a given method exceeds a median RFP score threshold. Each series corresponds to one of the homology detection methods described in the text.

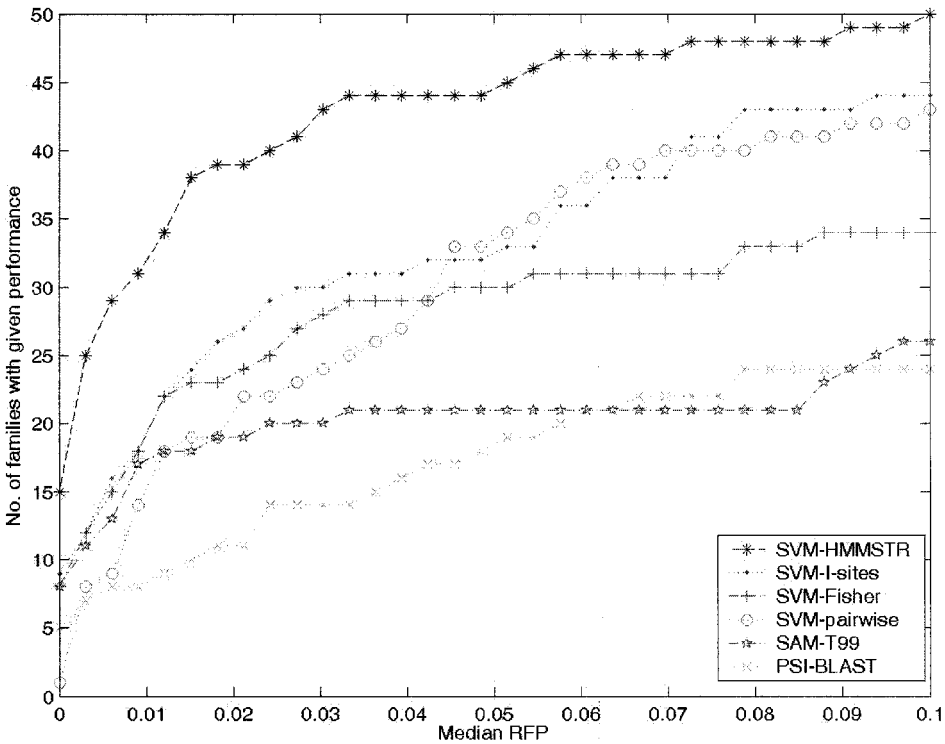

Fig. 7. Detail plot of the low median RFP region of Figure 6.

TABLE III. Statistical Significance of Differences Between Pairs of Homology Detection Methods

	SVM-HMMSTR	SVM-I-sites	SVM-Fisher	SVM-pairwise	SAM	PSI-BLAST
SVM-HMMSTR		1.35e-02	2.25e-02	0.0	4.5e-03	0.0
SVM-I-sites						9.0e-03
SVM-Fisher						3.15e-02
SVM-pairwise						1.35e-04
SAM						
PSI-BLAST						

Each entry in the table is the p value given by a Wilcoxon signed rank test comparing paired ROC50 scores from two methods for each of the 54 families. The p values have been adjusted for multiple comparisons using a Bonferonni adjustment. An entry in the table indicates that the method listed in the current row performs significantly better than the method listed in the current column at a threshold of 0.05.

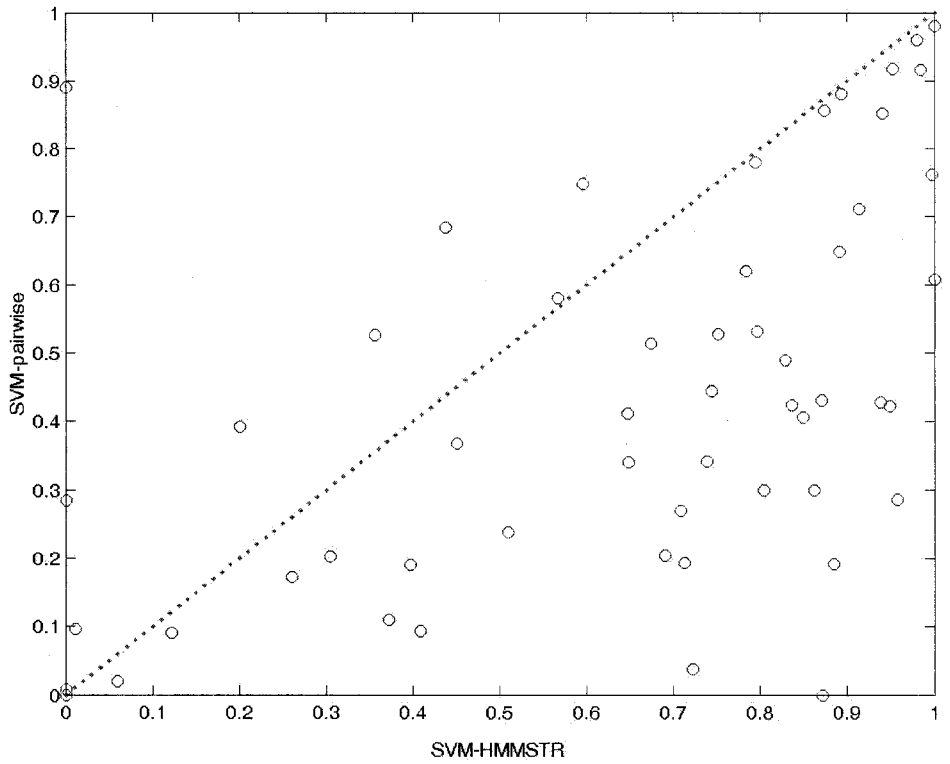

Fig. 8. Family-by-family comparison of SVMHMMSTR and SVM-pairwise. The coordinates of each point in the plot are the ROC50 scores for one SCOP family, obtained using SVM-HMMSTR and SVM-pairwise. The dotted line is $y = x$.

tion of these features, and it shows that HMMSTR does provide better features than the sequence-based models.

One significant characteristic of any homology detection algorithm is its computational efficiency. In this aspect, SVM-HMMSTR has the same order of time complexity as SVM-pairwise in theory. Both algorithms include an SVM optimization, which dominates SVM training time and is roughly $O(n^2)$,[11] where n is the number of training set examples. The vectorization step of SVM-pairwise involves computing n^2 pairwise scores. Using Smith–Waterman, each computation takes $O(m^2)$, yielding a total running time of $O(n^2m^2)$, where m is the length of the longest training set sequence. In contrast, SVM-HMMSTR first runs PSI-BLAST to obtain a profile before it aligns the obtained profile against HMMSTR to get the γ matrix. The time complexity of running PSI-BLAST on the Swissprot database is $O(N)$ when the length of the query sequence k is much less than N, where N is the size of the Swissprot database. Hence, the total running time of getting the profile is $O(nN)$. The size of Swissprot database N we used in the experiment is about quadratic in $m^* n$, where m is a second-order number; we conclude that the time complexity of the step of obtaining a profile is $O(n^2m^2)$. The step of aligning the obtained profiles for the n examples against HMMSTR is $O(nmp)$, where p is the number of HMM parameters. Thus, assuming that $m \approx p$, the time complexity of aligning a profile against HMMSTR is $O(nm^2)$. For feature representation and extraction step, the time complexity of obtaining order-independent feature set is an order of $O(nmp)$. The time complexity of obtaining order dependent feature set is $O(n^2m^2)$, because the time of computing the similarity score of 2 columns is a constant time for a small fixed number C. So the overall time complexity of SVM-HMMSTR is $O(n^2m^2)$. Therefore, we

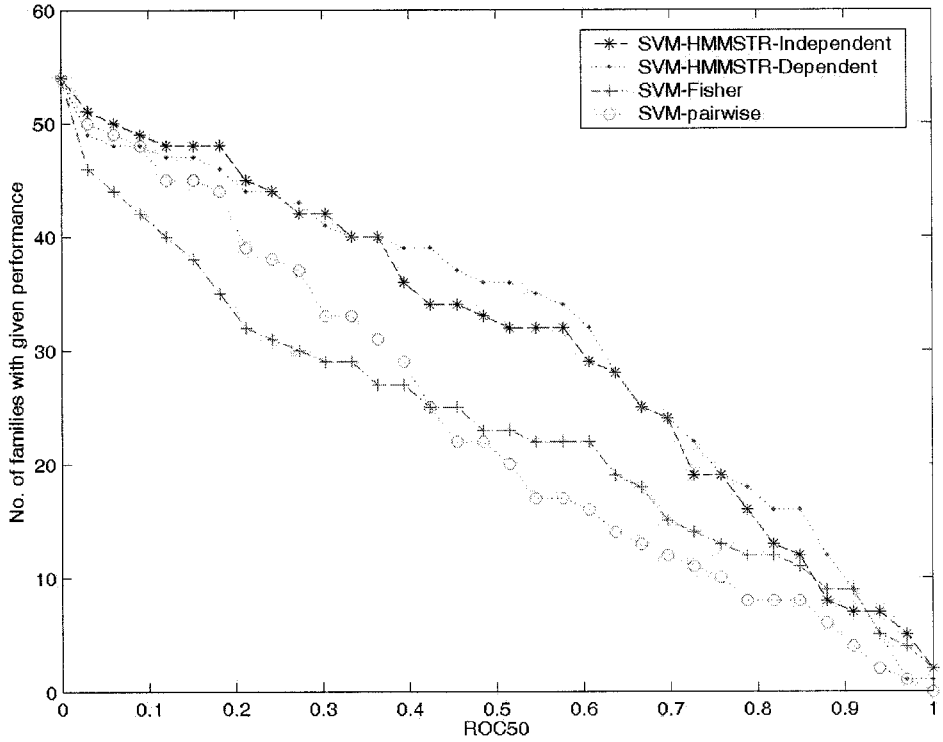

Fig. 9. Relative performance of homology detection methods. The graph plots the total number of families for which a given method exceeds a ROC50 score threshold. Each series corresponds to one of the homology detection methods described in the text.

conclude that the time complexity of SVM-HMMSTR is the same as that of SVM-pairwise.

CONCLUSIONS

To the best of our knowledge, this is the first attempt to encode both structure prediction and database alignment information in the training of an SVM. We have presented a new method to represent the sequence in terms of HMM state composition, and also in terms of alignment scores for sequences represented as HMM states. This creates a feature space that captures both the local structural composition and the overall conserved sequence similarity.

The improved performance of SVM-HMMSTR in remote homology detection can be understood by considering the following factors:

1. HMMSTR is a better model for local structure prediction than I-sites. HMMSTR states represent local sequence patterns, allowing distant similarities to be seen.
2. Our approach considers both the overall composition and the sequential ordering of local structures in separate feature sets. The compositional feature sets are sensitive but relatively unselective, while the sequential features are more selective but less sensitive, since they require alignments that may be error prone.
3. The use of SVMs enables efficient and effective learning to take place in high-dimensional feature space. The SVM owes a great part of its success to its ability to use kernels, allowing the classifier to exploit a very high-dimensional (possibly even infinite-dimensional) feature space. In addition to their empirical success, SVMs are also appealing due to the existence of strong generalization guarantees, derived from the margin-maximizing properties of the learning algorithm.

ACKNOWLEDGMENTS

We thank William Stafford Noble and Li Liao for their helpful discussions and for providing experimental results from their prior work, and Mark Diekhans for providing information about the Fisher kernel. We also thank the anonymous reviewers for their helpful comments and suggestions.

REFERENCES

1. Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 1970;48:443–453.
2. Smith T, Waterman M. Identification of common molecular subsequences. J Mol Biol 1981;147:195–197.
3. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. A basic local alignment search tool. J Mol Biol 1990;215:403–410.
4. Pearson WR. Rapid and sensitive sequence comparisons with FASTP and FASTA. Methods Enzymol 1985;183:63–98.
5. Gribskov M, McLachlan AD, Eisenberg D. Profile analysis: detection of distantly related proteins. Proc.Natl Acad Sci USA 1987;84: 4355–4358.
6. Krogh A, Brown M, Mian IS, et al. Hidden Markov models in computational biology: applications to protein modeling. J Mol Biol 1994;235:1501–1531.

7. Baldi P, Chauvin Y, Hunkapiller T, McClure MA. Hidden Markov models of biological primary sequence information. Proc Natl Acad Sci USA 1994;91:1059–1063.
8. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman OJ. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. Nucleic Acids Res 1997;25:3389–3402.
9. Karplus K, Barrett C, Hughey R. Hidden Markov models for detecting remote protein homologies. Bioinformatics 1998;14:846–856.
10. Jaakkola T, Diekhans M, Haussler D. A discriminative framework for detecting remote protein homologies. J Comp Biol 2000;7:95–114.
11. Liao L, Noble WS. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. J Comp Biol 2003;10:857–868.
12. Tsuda K, Kin T, Asai K. Marginalized kernels for biological sequences. The 10th International Conference on Intelligent Systems for Molecular Biology, 2002.
13. Ben-Hur A, Brutlag D. Remote homology detection: a motif based approach. The 11th International Conference on Intelligent Systems for Molecular Biology, 2003.
14. Leslie CS, Eskin E, Cohen A, Weston J, Noble WS. Mismatch string kernels for discriminative protein classification. Bioinformatics 2004;20:467–476.
15. Cristiani N, Shawe-Taylor J. An introduction to support vector machines. Cambridge, UK: Cambridge University Press; 2000.
16. Hou Y, Hsu W, Lee ML, Bystroff C. Efficient remote homology detection using local structure. Bioinformatics 2003;19:2294–2301.
17. Bystroff C, Baker D. Prediction of local structure in proteins using a library of sequence–structure motifs. J Mol Biol 1998;281:565–577.
18. Bystroff C, Thorsson V, Baker D. HMMSTR: a hidden Markov model for local sequence–structure correlations in proteins. J Mol Biol 2000;301:173–190.
19. Rabiner L. A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE 1989;77:257–286.
20. Eddy S. Profile hidden Markov models. Bioinformatics 1998;14:755–763.
21. Logan B, Moreno P, Suzek B, Weng Z, Kasif S. A study of remote homology detection. Compaq and DEC Technical Reports, CRL-2001-5, 2001.
22. Murzin AG, Brenner SE, Hubbard T, Chothia C. SCOP: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol 1995;247:536–540.
23. Yona G, Levitt M. Within the twilight zone: a sensitive profile–profile comparison tool based on information theory. J Mol Biol 2002;315:1257–1275.
24. Gribskov M, Robinson NL. Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. Comput Chem 1996;20:25–33.
25. Waterman MS, Vingron M. Sequence comparison significance and possion approximation. Stat Sci 1994;9:367–381.
26. Gumbel EJ. Statistics of extremes. New York: Columbia Univeristy Press; 1958.
27. Vapnik V. Statistical learning theory. New York: Wiley; 1998.
28. Park J, Karplus K, Barrett C, et al. Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. J Mol Biol 1998;284:1201–1210.