Efficient Remote Homology Detection Using Local Structure

Yuna Hou[1], Wynne Hsu[1], Mong Li Lee[1], and Christopher Bystroff[2]
[1]School of Computing,National University of Singapore,Singapore 117543
[2]Department of Biology,Rensselaer Polytechnic Institute,Troy, NY 12180, USA

ABSTRACT

Motivation: The function of an unknown biological sequence can often be accurately inferred if we are able to map this unknown sequence to its corresponding homologous family. At present, discriminative methods such as SVM-Fisher and SVM-pairwise, which combine support vector machine and sequence similarity, are recognized as the most accurate methods, with SVM-pairwise being the most accurate. However, these methods typically encode sequence information into their feature vectors and ignore the structure information. They are also computationally inefficient. Based on these observations, we present an alternative method for SVM-based protein classification. Our proposed method, SVM-I-sites, utilizes structure similarity for remote homology detection.
Result: We run experiments on the SCOP 1.53 dataset. The results show that SVM-I-sites is more efficient than SVM-pairwise. Further, we find that SVM-I-sites outperforms sequence-based methods such as PSI-BLAST, SAM, and SVM-Fisher while achieving a comparable performance with SVM-pairwise.
Availability: I-sites server is accessible through the web at http://www.bioinfo.rpi.edu. Programs are available upon request for academics. Licensing agreements are available for commercial interests. The framework of encoding local structure into feature vector is available upon request.
Contact:houyuna@comp.nus.edu.sg, bystrc@rpi.edu

1. INTRODUCTION

Proper identification of homologous relationships in proteins is important in advancing our understanding of the functions of biological sequences. While the amount of discovered biological sequences has increased at an unprecedented pace, the rate of analyzing, mapping, and understanding these sequences remains unacceptably slow. As a result, molecular biologists are turning to computational techniques to help the analysis of these data.

Much research has been focused on protein homology detection. Dynamic programming based alignment tools such as Smith-Waterman (Smith *et al.*, 1981) and their efficient approximations such as BLAST (Altschul *et al*, 1990) and FASTA (Pearson, 1985) have been widely used to provide evidence for homology by matching a new sequence against a database of previously annotated sequences. However, these approaches can only detect homologous proteins that exhibit significant sequence similarity. In order to detect weak or remote homologies, one can utilize the concept of protein family or superfamily, which denotes a group of sequences sharing the same evolutionary origin.

A statistical model can also be built for each family or superfamily, and a new sequence is subsequently compared with these models. In contrast to simple pairwise comparison methods, the ability to match a sequence to superfamily-based models computationally often allow the biologists to infer nearly three times as many homologies (Park *et al.*, 1998). Profiles (Gribskov *et al.*, 1987) and hidden Markov models (Krogh *et al.*, 1994; Baldi, *et al.*, 1994) are two methods commonly used for representing these models. These probabilistic models are often called generative because the methodology involves building a model for a single protein family and then evaluating each candidate sequence to see how well it fits the model. If the “fit” is above some threshold, then the protein is classified as belonging to the family.

By gleaning the extra information of unlabeled protein sequences in large databases, iterative methods such as PSI-BLAST (Altschul *et al.*, 1997) and SAM (Karplus *et al.*, 1998) improve upon profile-based methods by iteratively collecting homologous sequences from a large database and incorporating the resulting statistics into a central model.

A recently proposed approach called the discriminative method is able to attain additional accuracy by modelling the difference between positive and negative examples explicitly. There are two steps in this approach: a given set of proteins is first converted into fixed-length vectors, before an SVM is trained from the vectorized proteins. The most prominent works that employ this approach include SVM-Fisher (Jaakkola *et al.*, 2000) and SVM-pairwise (Liao *et al.*). The two methods differ mainly in the vectorization step. In SVM-Fisher, a protein’s vector representation is its gradient with respect to a profile hidden Markov model; while in SVM-pairwise, the vector is a list of pairwise sequence similarity scores. While the SVM-pairwise method is currently the most accurate method for detecting remote homologies, it is inefficient and not scalable.

All the above works detect remote homology using only sequence information. This is accurate only if the proteins are closely related. In this paper, we offer an efficient vectorization method while maintaining a comparable performance with SVM-pairwise. We observe that the three-dimensional structures of a set of homologous proteins are conserved to a greater extent than their primary sequences. Therefore, we encode structure information into feature vectors instead of using sequence similarity for remote homology detection. Here, we assume that the structure information is given by the probability that the protein contains certain local structure, as predicted by a library of sequence-structure motifs I-sites library (Bystroff *et al.*, 1998). Experimental results on SCOP1.53 databases demonstrate that the accuracy of our proposed method is comparable with the state-of-the-art method SVM-pairwise and outperforms methods such as PSI-BLAST, SAM and SVM-Fisher.

2. SYSTEM AND METHODS

2.1 Overview

Figure 1 gives the overview of the proposed SVM-I-sites method. It consists of two phases: (a) the training phase which constructs support vector classifiers, and (b) the testing phase which uses a support vector machine (SVM) to determine if the protein belongs to some known protein classes. Both phases require the extraction of features from the proteins and represent them in some suitable form, which essentially distinguishes our method from SVM-Fisher and SVM-pairwise.

In SVM-Fisher, a protein's vector representation is its gradient with respect to profile hidden Markov model. On the other hand, SVM-pairwise method uses a pairwise sequence similarity algorithm Smith-Waterman in place of the HMM in the SVM-Fisher method. Both SVM-Fisher and SVM-pairwise methods ignore the structure information when encoding the feature vector. In SVM-I-sites, we encode the local structure information into the feature vector. By doing this, we incorporate a natural biological interpretation into our method to capture parts of the "signature" of the protein's three-dimension structure.

During the training phase, the proteins in the database is transformed into high-dimensional feature vectors. These feature vectors are separated into two classes: the positive examples (which refer to those feature vectors that belong to the protein classes) and the negative examples (which refer to those feature vectors that do not belong to the known protein classes). An SVM is subsequently constructed to discriminate the positive and negative examples. This process is repeated for all protein classes under investigation. The output from the training phase is a set of SVM, one for each protein class.

In the testing phase, a high-dimensional feature vector is created for the protein under investigation. Each of the trained SVM is then queried to determine whether the given protein belongs to the particular protein class associated with the SVM. A positive result would suggest that the protein under investigation has a homologous relationship with the corresponding protein class.

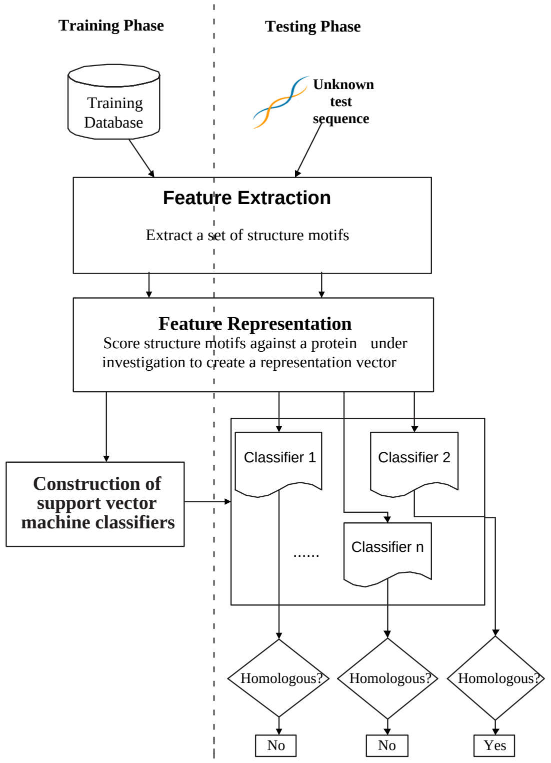


Figure 1: Overview of SVM-I-sites

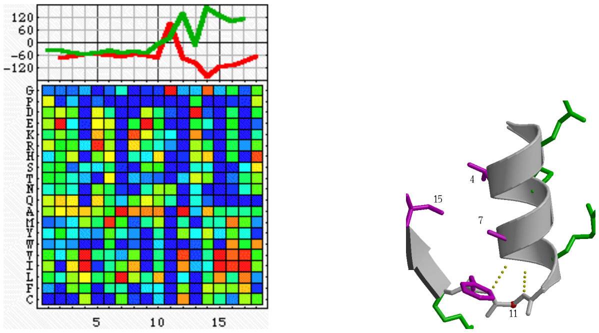
2.2 Feature extraction and representation

While sequence information does provide important hints to the presence of homologous relationship, it is not sufficient. In fact, there exists a large number of proteins that are homologous but whose sequences are only weakly similar. For these remote homologous proteins, we observe that their three-dimensional structures share many common characteristics. Thus, it would be useful to capture these common structures and represent them in a form suitable for the subsequent training and testing of support vector machine algorithms.

The most straightforward way to incorporate structure information in remote homology detection is to encode them into the features. Unfortunately, three-dimensional protein structures cannot be accurately predicted from sequences. An intermediate but useful step is to predict the protein secondary structure, by projecting the complicated three-dimensional structure onto one dimension, i.e. onto a string of secondary structural assignments for each residue. Sequences which are distantly related to each other but which have similar functions, tend to have highly conserved patterns of secondary structure (Russell *et al.*, 1994). A better 1D representation of proteins is the generalized "local structure", which includes two of the three secondary structure types (helix and strand) but reclassifies the loop states to one of several different loop types, such as the Schellman cap motif shown in Figure 2. These loop motifs often have specific sequence signatures that are conserved between remote homologs.

Pioneering work in protein secondary structure prediction includes (Efimov, 1993; Hutchinson *et al.*, 1994; Zhu *et al.*, 1996; Oliva *et al.*, 1997; Han *et al.*, 1996). However, the majority of these methods do not identify the strong relationship between the amino acid sequence and structure. Further, the focus of these methods is on the three-state secondary structure prediction, namely helix, stand and loop. Hence, they are not suitable for encoding secondary structure information into feature vectors.

One of the most successful local structure prediction methods is by Bystroff and Baker (Bystroff *et al.*, 1998), which performs local structure prediction based on a library (I-sites library) of short sequence patterns (profiles) that correlate strongly with protein three-dimensional structure elements. In the I-sites library, there are 263 sequence-structure profiles each of which corresponds to a unique structure motif which are more specific than the three-state secondary structure. Figure 2 is an example of sequence-structure profile.



(a)Sequence pattern (b)Glycine alpha-C-cap Type 1

Figure 2: This is one of the sequence profiles and its corresponding local structure in I-sites library. (a) is the sequence pattern for Glycine alpha-C-cap Type 1. Along the Y-axis are the 20 amino acids, arranged roughly from non-polar on the bottom to polar on the top, except that glycine and proline are on the top and cystine is on the bottom. Along the X-axis is the position in the motif, each column represents one amino acid. The different color represents frequency of occurrence: red for high frequency (log-odds ratio $>$ 3), and blue for low frequency (log-odds ratio $<$ -3). (b) is the three-dimension element which has a strong correlations with the sequence patterns of (a). In the Type 1 glycine cap, an amphipathic helix is followed immediately by a glycine and an aspartate beta-bend. The aspartate is preferred in the position two residues after the glycine. Conserved non-polar sidechains 1 and 4 residues after the glycine interact with two conserved non-polar sidechains 4 and 7 residues before the glycine.

In order to predict the local structure of any unknown protein sequence, the sequence patterns (profiles) for each of the 263 clusters of I-sites library are used to score all subfragments of this unknown target sequence. Given the length difference, the similarity scores of different clusters are not directly comparable. Instead the associated "confidence" values are compared. The confidence of a fragment prediction is the probability that a sequence segment with a given score has the predicted structure. Each cluster has a confidence curve (Figure 3) which maps similarity score to the probability of correct local structure based on a ten-fold jack-knife test: 90% of the database is used to refine each of the cluster, while the remaining 10% is set aside for scoring the profile of this cluster. The top-scoring segments are kept. The structures of the top-scoring segments are then compared to the paradigm structure for the cluster, chosen from the 90% training set. The list is sorted by score and the fraction of true-positive in the cluster. This procedure is repeated ten times using a different 10% as the test set and the results are averaged. Please refer to (Bystroff *et al.*, 1998) for the details of the confidence curve generation procedure.

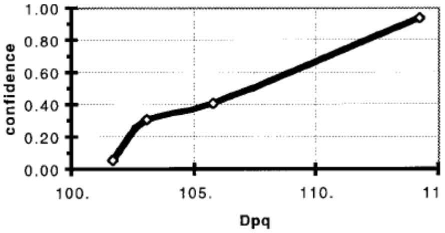


Figure 3: A confidence curve map similarity score to confidence.

Given any protein sequence, we use the following method to obtain its structure features: the first step is to run PSI-Blast against swissprot database to generate a multiple sequence alignment. After that, the multiple alignment is converted to a sequence profile, and the sequence profile is searched in overlapping windows using each of the 263 I-sites and we get a score for each sub-fragment. Finally, each score is translated into a probability ("confidence") value, and the whole set of "confidence" values are sorted. Thus, for each sub-fragment, we obtain the probability ("confidence" value) of this subsequence belonging to each of the 263 structure motifs.

To minimize the effect due to mutation, we apply a threshold such that if the "confidence" value falls below the threshold, then the confidence of this subsequence will be set to zero. The number 0.25 is the default value of I-sites program. Initial experiments show that using a threshold of 0.5 gives the best performance. Details of the experiments are given in Section 4.4.

Several heuristics can be used to handle the situation where a protein motif occurs multiple times. For example, we can take the maximum, the sum, or the average of all the "confidence" value for such protein motifs. Initial experiments to determine a good heuristics show that using the sum value gives the best performance. Details of the experiment are given in Section 4.4. Table 1 shows a sample of the vector generated for the protein d9atcb2. Each value in the vector denotes the sum "confidence" value of the corresponding local structure occurring in the protein.

It is possible that two(or more) overlapping subsequences show similarity to different I-sites motifs. At the same time, many of I-sites motifs tend to overlap. This may cause one single subsequence to yield high confidence value to multiple different motifs. In this paper, we keep all the predictions

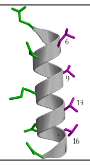
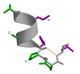
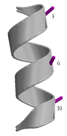
Local Structure	Feature Value
	4.76
	3.84
	4.23
...	...

Table 1: A sample of the generated structure feature values

even when there exists some overlap. In future, we shall look into some appropriate overlap removing algorithm to improve the performance.

2.3 Construction of SVM classifiers

Having obtained the feature vectors for the proteins, the next step is to predict whether the given feature vector exhibits homologous relationship with any of the known protein families. Classical machine learning techniques such as Naive Bayes classifiers (Lanley *et al.*, 1992), neural networks (Pao, 1989), decision tree classifiers (Quinlan, 1993) etc do not perform well for remote homology detection because they are unable to effectively obtain good generalization from sparse training data in high dimensions.

The well-established SVM exhibits excellent generalization performance in practice and is grounded in statistical learning theory (Vapnik, 1998). The idea behind SVM is to locate a hyperplane that maximizes the distance separation between the positive and negative examples. Figure 4 shows the two-dimensional case. Three possible lines are drawn to separate the positive and negative examples. The highlighted line is the one chosen by SVM since it maximizes the distance separation between the positive and negative examples.

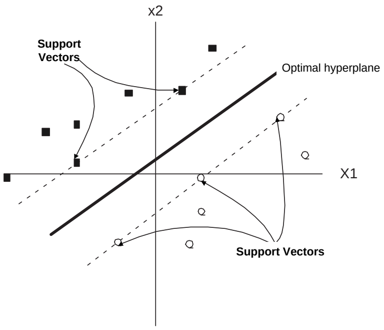


Figure 4: Support Vector Machines

We first train the SVM to find such a partitioning hyperplane. Then the SVM predict the classification of an unknown protein by mapping it into the feature space and determining which side of the hyperplane the unknown protein lie on.

In our implementation, we use the *gist* SVM software implemented by Noble and Pavlidis (Noble *et al.*). It contains a kernel function that acts as the similarity score between pairs of input vectors. The base kernel is normalized so that each vector has length 1 in the feature space; i.e.,

$$K(X, Y) = \frac{X \cdot Y}{\sqrt{(X \cdot X)(Y \cdot Y)}}$$

The parameters needed to tune a SVM are the 'capacity' and the choice of kernel. The capacity allows us to control how much tolerance we allow for errors in the classification of training samples. This affects the generalization ability of the SVM and prevents overfitting. In our experiments, we use a capacity equals to 10, which guarantees the numerical stability of the SVM algorithm yet provides sufficient generalization.

The kernel function allows the SVM to create hyperplanes in high dimensional spaces that effectively separate the training data. In the input space, training vectors are often not separated by a simple hyperplane. The kernel maps data from one space to another such that a simple hyperplane can effectively separate the data into two classes. We employ the Gaussian kernel for all the classifiers. The variance of the associated Gaussian Kernel is computed as the median Euclidean distance (in feature space) from any positive training examples to the nearest negative example. The output is a discriminant score that is used to rank the members of the test set.

To determine whether an unlabelled protein belongs to a particular protein class, we test it against the SVM trained for that class. The SVM classifier produces a 'score' representing the distance of the the testing feature vector from the margin. The larger the score is, the further away the vector is from the margin, and the more confident we are of the classifier's prediction.

3. COST ANALYSIS

Computational efficiency is an important characteristic for any homology detection algorithm. In this respect, the SVM-I-sites method is more efficient than SVM-pairwise. Both SVM-I-sites and SVM-pairwise include an SVM optimization and vectorization step. In the optimization step, both algorithms take $\mathcal{O}(n^2)$ time, 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 $\mathcal{O}(m^2)$, yielding a total running time of $\mathcal{O}(n^2 m^2)$, where m is the length of the longest training set sequence.

In contrast, SVM-I-sites first runs PSI-BLAST to obtain a profile before it runs I-sites function to compute the "confidence" value of each sequence containing a pre-defined local structure. 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. The time complexity of running I-sites function is $O(k)$ for each sequence. Hence, the total running time of SVM-I-sites is $O(nN)$. Since N is typically one order of $m * n$ where m is a two order number, we conclude that SVM-I-sites is about one order faster than SVM-pairwise.

We also carry out an experiment to compare the response time of SVM-pairwise and SVM-I-sites for the vectorization step. The time taken by SVM-pairwise includes the CPU time and the output time of the n^2 pairwise scores, while the time taken by SVM-I-sites includes the CPU time and output time of PSI-BLAST and I-sites function. This experiment is performed on a Pentium III 750MHz Ultra Sparc running SunOS 5.8 with 8 GB RAM. I-sites takes 19 hours, and the Smith-Waterman algorithm requires 70 hours. Clearly, SVM-I-sites is 4 times faster than SVM-pairwise in vectorization step. Note that the real-time comparison is not as significant as the theoretical analysis because the number of I/Os incurred by SVM-I-sites is much more than that for SVM-pairwise.

4. PERFORMANCE STUDY

In this section, we compare the performance of five algorithms: SVM-I-sites, PSI-BLAST, SAM, SVM-Fisher and SVM-pairwise.

4.1 Experiment setup

We evaluate the accuracy of each algorithm by its ability to classify protein domains into superfamilies in the Structural Classification of Proteins (SCOP)(Murzin *et al.*, 1995) version 1.53. Sequences are selected using the Astral database (astral.stanford.edu (Brenner *et al.*, 2000)), and similar sequences are removed using an E-value threshold of 10^{-25}. This procedure resulted in 4352 distinct sequences, grouped into families and superfamilies. The database is selected so as to provide a direct comparison with previous work on remote homology detection method, namely, SVM-pairwise.

We use the same experiment setup as SVM-pairwise. For each family, the protein domains within a family are considered positive test examples; while the protein domains outside the family but within the same superfamily are taken as positive training examples. The data set yields 54 families containing at least 5 family members (positive test) and 10 superfamily members outside of the family (positive train). 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.

4.2 Comparative methods

The vectorization step of SVM-pairwise uses the Smith-Waterman algorithm as implemented on the BioXLP hardware accelerator (www.cgen.com). The feature vector corresponding to protein X is $F_X = f_{x1}, f_{x2}, ..., f_{xn}$, where n is the total number of proteins in the training set and f_{xi} is the E-value of the Smith-Waterman score between sequence X and the ith training set sequence. The default parameters are used: gap opening penalty and extension penalties of 11 and 1, respectively, and the BLOSUM 62 matrix.

For comparison, we also include the result of PSI-BLAST, SAM and SVM-Fisher methods presented in the SVM-pairwise paper (Liao *et al.*).

In SAM experiment, the Hidden Markov models are trained using the Sequence Alignment and Modeling (SAM) toolkit (www.soe.ucsc.edu/research/compbio/sam.html)(Krogh *et al.*, 1994). Models are built from unaligned positive training set sequences using the local scoring option ("-SW 2"). Following (Jaakkola *et al.*, 2000), a 9-component Dirichlet mixture prior developed by Kevin Karplus (byst-4.5-0-3.9comp at www.soe.ucsc.edu/research/compbio/dirichlets) is used. After the model is built, each of the test sequences is compared to the model by using *hmmscore*(also with the local scoring option) and the resulting E-values are used to rank the test set sequences.

The SVM-Fisher method uses the same trained HMMs during the vectorization step. Then, the forward and backward matrices are combined to yield an observation count for each parameter in the HMM. As shown in (Jaakkola *et al.*, 2000), the counts can be converted into components of a gradient vector. Although these gradient components can be computed for every HMM parameter, the SVM-Fisher method uses only the gradient components that correspond to emission probabilities in the match states. Furthermore, a more compact gradient vector can be derived using a mixture decomposition of the emission probabilities. For a profile HMM containing m match states, the length of the resulting vector is $9m$.

In PSI-BLAST, the input is a single sequence, whereas for methods such as HMMER and SVM-Fisher, the input consists of multiple input sequences. In our experiments, we randomly select a positive training set sequence to serve as the initial query. The complete positive training set is aligned using CLUSTALW (Thompson *et al.*, 1994). Then the query sequence and the alignment is used as inputs. We run one iteration of PSI-BLAST with the test set as a database. Note that PSI-BLAST is not run on the test set for multiple iterations: this restriction allows a fair comparison with the other, non-iterative methods included in the study. The resulting E-values are used to rank the test set sequences.

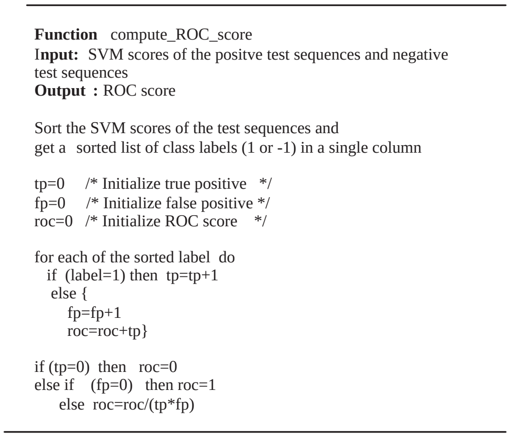
Note that the setup of SAM and PSI-BLAST methods as presented in SVM-pairwise is slightly different from their commonly reported usuage (Park *et al.*, 1998). It is possible for SAM, PSI-BLAST and SVM-Fisher to achieve better performance than those reported here if they have the benefit of putative homologs from large sequence databases as (Park *et al.*, 1998).

4.3 Performance metrics

We use the two scoring methods as reported in SVM-pairwise (Liao *et al* to compare these methods:

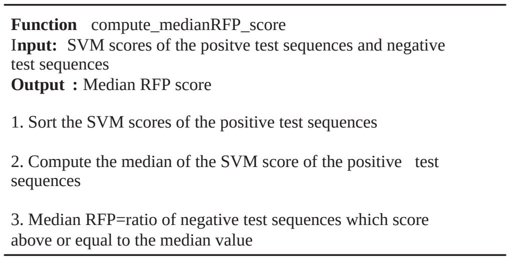
1. Receiver operating characteristic(ROC) scores.
2. Median rate of false positives(RFP).

The ROC score is the area under the receiver operating characteristic curve – the plot of true positives as a function of false positives (Gribskov *et al.*, 1996). A score of 1 indicates perfect separation of positives from negatives, whereas a score of 0 denotes that none of the sequences selected by



```
Function   compute_ROC_score
Input:  SVM scores of the positve test sequences and negative
test sequences
Output : ROC score

Sort the SVM scores of the test sequences and
get a  sorted list of class labels (1 or -1) in a single column

tp=0    /* Initialize true positive   */
fp=0    /* Initialize false positive */
roc=0   /* Initialize ROC score    */

for each of the sorted label  do
   if  (label=1) then  tp=tp+1
    else {
        fp=fp+1
        roc=roc+tp}

if (tp=0)  then   roc=0
else if    (fp=0)   then roc=1
      else  roc=roc/(tp*fp)
```

Figure 5: Algorithm to compute ROC score



```
Function   compute_medianRFP_score
Input:  SVM scores of the positve test sequences and negative
test sequences
Output : Median RFP score

1. Sort the SVM scores of the positive test sequences

2. Compute the median of the SVM score of the positive   test
sequences

3. Median RFP=ratio of negative test sequences which score
above or equal to the median value
```

Figure 6: Algorithm to compute median RFP score

the algorithm is positive. The algorithm to compute ROC score is shown in Figure 5. The median RFP score is the fraction of negative test sequences that score as high or better than the median-scoring positive test sequence. The algorithm to compute median RFP score is shown in Figure 6.

4.4 Experiment results

Table 2 summarizes the average ROC score for the 54 SCOP families of using different heuristics to account for multiple subsequence occurrences as described in Section 2.2 at default threshold 0.25. Table 2 shows that the sum heuristics gives the best performance.

Heuristic methods	Average ROC score for the 54 SCOP families
Maximum	0.88
Sum	0.89
Average	0.86

Table 2: Results of the experiments to determine the best heuristics

Table 3 summarizes the average ROC score for the 54 SCOP families of using different thresholds and sum to account for multiple subsequence occurrences. Table 3 shows that it gives the best performance at threshold 0.5. Here, the final reported performance is the combination of 0.5 threshold and sum to account for multiple subsequence occurrences.

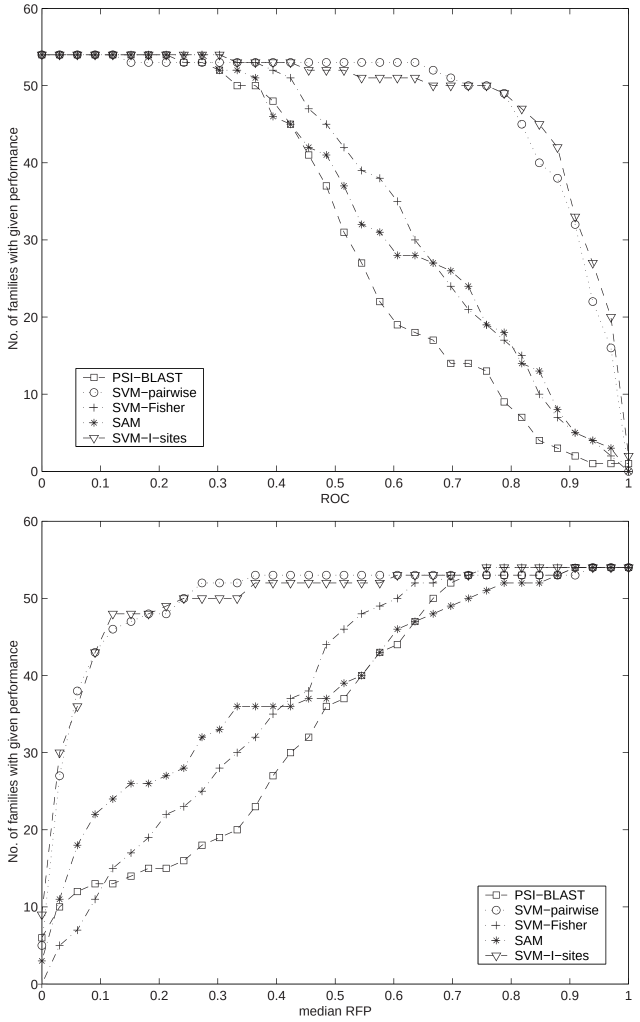


Figure 7: Relative performance of homology detection methods. Each graph plots the total number of families for which a given method exceeds a score threshold. The top graph uses ROC scores, and the bottom graph uses median RFP scores. Each series corresponds to one protein homology detection methods.

Thresholds	Average ROC score for the 54 SCOP families
0.25	0.89
0.4	0.89
0.5	0.90
0.6	0.88

Table 3: Results of the experiments to determine the best threshold

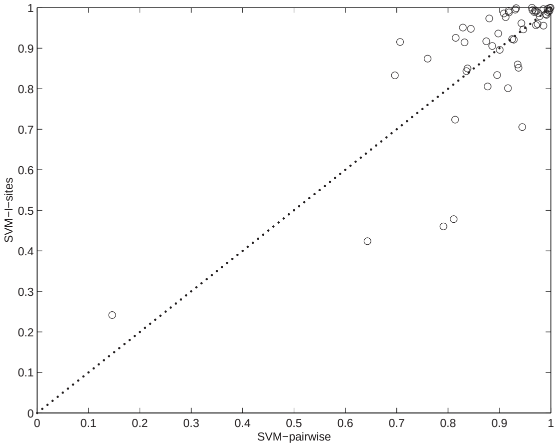


Figure 8: Family-by-family comparison of SVM-pairwise and SVM-I-sites. Each point on the graph corresponds to one of the 54 SCOP superfamilies. The axes are ROC scores achieved by the two primary methods compared in this study: SVM-pairwise and SVM-I-sites

The results of the comparative experiment are summarized in Figure 7. The two graphs rank the five homology detection methods according to ROC and median RFP scores. In each graph, a higher curve corresponds to more accurate homology detection performance. We observe that SVM-I-sites performs significantly better than PSI-BLAST, SAM and SVM-Fisher methods, and is comparable to SVM-pairwise. This is because SVM-pairwise adopt pairwise scores as feature values and the Smith-Waterman algorithm is recognized as the most sensitive pairwise comparison method.

SVM-I-sites can be an alternative and complimentary method to SVM-pairwise method to construct features with local structure probabilities. This can be shown from Figure 7. Figure 8 is a family-by-family comparison of the 54 ROC scores computed for each method. The results suggest that SVM-I-sites and SVM-pairwise are two complimentary methods for detection remote homology.

5. DISCUSSION

The inference of homology relationship in proteins with known structure and/or function is a core problem in computational biology. Sequence comparison is the most commonly used approach to determine homology. However, remote homologous proteins tend to have little sequence similarities. As such, they are often statistically undetectable using conventional sequence comparison methods. Homology or common ancestry in such cases needs to be inferred from their common three-dimensional structures and functions.

The main novelty of our work is in investigating how local structure information can help remote homology detection. By using local structure features, we seek to develop an approach that has a natural biological interpretation. Further, we have described an integrated framework to construct feature vectors that encode structure information. The local structure is encoded into the feature vector so that parts of the three-dimension "signature" is captured. The use of support vector machines also enables learning to take place in high dimensional feature space. Our experiment results confirm that it is important to incorporate structure information in the feature space.

Efficiency is another advantage of SVM-I-sites compared to SVM-pairwise. SVM-I-sites is more efficient in the vectorization step, thus making it a more practical solution for large databases.

In addition, SVM-I-sites method shares many advantages as SVM-pairwise. First, it does away with the need for profile HMM topology and parameterization. Second, it learns from both positive and negative training examples, while a profile method is trained solely on a collection of positive examples. Third, it does not require a multiple alignment of the training set sequence which may not be possible for distantly related protein sequences.

Current work ignores the local structure order. This may result in proteins containing the same local structure but with different orders being classified into the same superfamily. Ongoing work includes investigating how the local structure order influence the remote homology detection performance.

6. ACKNOWLEDGMENTS

We thank William Stafford Noble and Li Liao for their helpful discussion and for providing the experiment results from their prior work.

7. REFERENCES

[1] Altschul,S. F., Gish,W., Miller,W., Myers,E. W., and Lipman, D. J. (1990) A basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410.

[2] Altschul,S. F., Madden,T. L., Schaffer,A. A., Zhang,J., Zhang,Z., Miller,W., and Lipman,D. J. (1997) Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402.

[3] Baldi,P., Chauvin,Y., Hunkapiller,T., and McClure,M. A. (1994) Hidden Markov models of biological primary sequence information. *PNAS*, 91(3):1059–1063.

[4] Brenner,S. E., Koehl,P., and Levitt,M. (2000) The astral compendium for sequence and structure analysis. *Nucleic Acids Research*, 28:254–256.

[5] Bystroff,C. and Baker,D. (1998) Prediction of local structure in proteins using a library of sequence-structure motifs. *J.Mol.Biol.*, 281:565–577.

[6] Efimov,A. V. (1993) Standard structures in proteins. *Prog. Biophys. Mol. Biol.*, 60:201–239.

[7] Gribskov,M., McLachlan,A., and Eisenberg,D. (1987) Profile analysis: Detection of distantly related proteins. *PNAS*, USA 84:4355–4358.

[8] Gribskov,M. and Robinson,N. L. (1996) Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Computer and Chemistry*, 20(1):25–33.

[9] Han,K. F. and Baker,D. (1996) Global properties of the mapping between local amino acid sequence and local structure in proteins. *PNAS*, USA 93:5814–5818.

[10] Hutchinson,E. G. and Thornton,J. M. (1994) A revised set of potentials for beta-turn formation in proteins. *Protein Sci.*, 3:2207–2216.

[11] Jaakkola,T., Diekhans,M., and Haussler,D. (2000) A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7(1-2):95–114.

[12] Karplus,K., Barrett,C., and Hughey,R. (1998) Hidden markov models for detecting remote protein homologies. *Bioinformatics*, 14(10):846–856.

[13] Krogh,A., Brown,M., Mian,I. S., Sjolander,K., and Haussler,D. (1994) Hidden markov models in computational biology: Applications to protein modeling. *JMB*, 235:1501–1531.

[14] Lanley,P., Iba,W., and Thompson,K. (1992) An analysis of bayesian classifiers. In *Proceedings of the tenth national conference on artificial intelligence*, pages 223–228. AAAI press and MIT press.

[15] Liao,L. and Noble,W. S. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. *Journal of Computational Biology*, To appear.

[16] Thompson,J.D., Higgins,D.G., Gibson,T.J. (1994) CLUSTALW: Improving the sensitivity of progressive multiple alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22):4673–4680.

[17] Murzin,A. G., Brenner,S. E., Hubbard,T., and Chothia,C. (1995) Scop: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247:536–540.

[18] Noble,W. S. and Pavlidis,P. www.cs.columbia.edu/compbio/svm.

[19] Oliva,B., Bates,P. A., Querol,E., Aviles,F. X., and Sternberg,M. J. E. (1997) An automated classification of the structure of protein loops. *Journal of Computational Biology*, 266:814–830.

[20] Pao,Y. (1989) *Adaptive Pattern Recognition and Neural Networks.* New York, NY: Addison Wesley.

[21] Park,J., Karplus,K., Barrett,C., Hughey,R., Haussler,D., Hubbard,T., and Chothia,C. (1998) Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *J.Mol.Biol.*, 284(4):1201–1210.

[22] Pearson,W. R. (1985) Rapid and sensitive sequence comparisons with FASTP and FASTA. *Methods in Enzymology*, 183:63–98.

[23] Quinlan,J. (1993) C4.5: Programs for machine learning. *Morgan Kaufmann.*

[24] Russell,R. B. and Barton,G. (1994) Structure features can be unconserved in proteins with similar folds. *J. Mol. Biol.*, 244:332–350.

[25] Smith,T. and Waterman,M. (1981) Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197.

[26] Vapnik,V. N. (1998) *Statistical Learning Theory.* Springer.

[27] Zhu,Z. Y. and Blundell,T. L. (1996) The use of amino acid patterns of classified hilices and strands in secondary structure prediction. *J. Mol. Biol.*, 260:261–276.