Efficient Remote Homology Detection with Secondary 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. Currently, discriminative approach which combines support vector machine and sequence similarity is recognized as the most accurate approach. SVM-Fisher and SVM-pairwise methods are two representatives of this approach, and SVM-pairwise is the most accurate method. However, these methods only encode sequence information into their feature vectors and ignore the structure information. In addition, one of their major drawbacks is their computation inefficiency. Based on this observation, we present an alternative method for SVM-based protein classification. Our method, SVM-I-sites, uses structure similarity instead of sequence similarity for remote homology detection. Our studies show that SVM-I-sites is much more efficient than both SVM-Fisher and SVM-pairwise while achieving a comparable performance with SVM-pairwise.
Result: We adopt SCOP 1.53 as our dataset. The result shows that SVM-I-sites runs much faster and is able to outperform many state-of-the-art sequence-based methods such as PSI-BLAST, SAM and SVM-Fisher, and comparable to 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, more and more molecular biologists have turned to computation methods to help in the analysis of these data.

Much research has been focused on protein homology detection. Dynamic programming based alignment tools such as Smith−Waterman [24] and their efficient approximations such as BLAST [1] and FASTA [21] 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. One can build a statistical model for each family or superfamily and then compare a new sequence to a collection of models. Computational method that relates a sequence to such superfamily-based models often allow the computational biologists to infer nearly three times as many homologies as simple pairwise comparison methods [20]. Profiles [7] and hidden Markov models [13, 3] are two methods 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 [2] and SAM [12] improve upon profile-based methods by iteratively collecting homologous sequences from a large database and incorporating the resulting statistics into a central model.

Most recently a new approach called discriminative method gains additional accuracy by modelling the difference between positive and negative examples explicitly. In this approach, there are two steps: converting a given set of proteins into fixed-length vectors, and training an SVM from the vectorized proteins. The most successful work in this approach includes SVM-Fisher [11] and SVM-pairwise [15]. The two methods differ only in the vectorization step. In the SVM-Fisher method, a protein's vector representation is its gradient with respect to a profile hidden Markov model; in the SVM-pairwise method, the vector is a list of pairwise sequence similarity scores. SVM-pairwise method is currently the most accurate method for detecting remote homologies.

One major drawback of SVM-pairwise method is its computational inefficiency. SVM-pairwise method is inefficient in the vectorization step. As the database grows quickly, the slow speed of SVM-pairwise will be impractical to use. In this paper, we provide a more efficient way of vectorization step while getting a comparable performance with SVM-pairwise.

In addition, all of these above works detect remote homology using only sequence information. Remote homology detection depends on sequence information can reveal homology accurately if the proteins are closely related. One important observation in remote homology detection is that for a set of proteins that are hypothesized to be homologous, their three-dimensional structures are conserved to a greater extent than are their primary sequences. Based on the above observation, we encode structure information into feature vectors instead of using sequence similarity for remote homology detection. In our work, structure information is indicated by the probability that the protein contains certain local structure, as predicted by a library of sequence-structure motifs I-sites library [5]. Experimental results on SCOP1.53 databases show that the accuracy 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 method. It consists of two phases: (a) the training phase which constructs support vector classifiers, and (b) the testing phase which use a support vector machine 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.

Hence, the critical issue in this general framework of homology detection lies in the feature extraction and representation. The difference between our work with SVM-Fisher and SVM-pairwise is the feature extraction and representation procedure. In SVM-Fisher method, a protein's vector representation is its gradient with respect to profile hidden Markov model; in SVM-pairwise method, the feature vector corresponding to a 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. Both of SVM-Fisher and SVM-pairwise methods are the most successful work in detecting remote homology. However, both of them ignore the structure information when encoding the feature vector. In our work, we encode the local structure information into the feature vector. By encoding the local structure information, we seek to develop an approach that has a natural biological interpretation which can capture parts of the "signature" of the three-dimension structure.

We create a feature vector for the protein under investigation by scoring a set of local structure motifs generated from the given protein. At the end of the scoring process, we obtain a high-dimensional feature vector corresponding to the protein under investigation.

During the training phase, we transform the proteins in

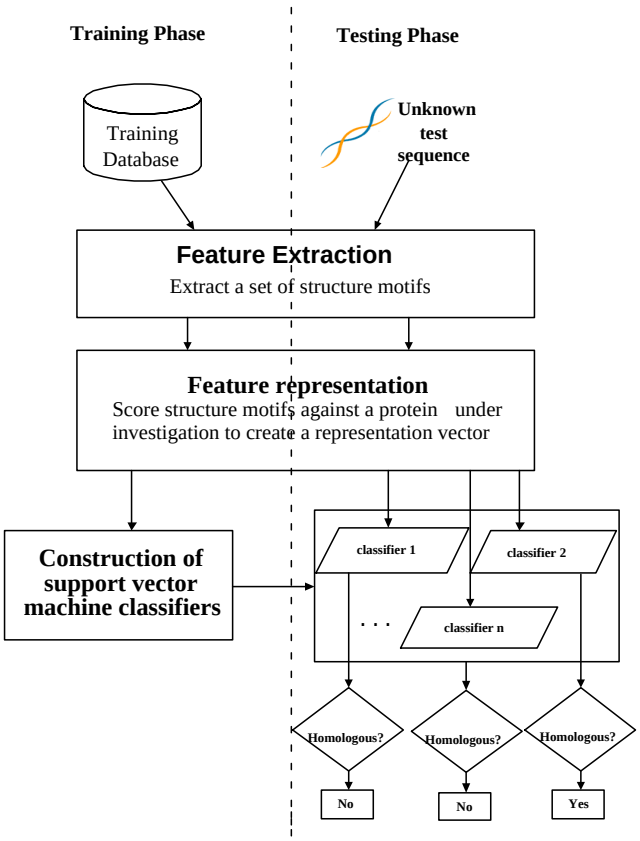


Figure 1: Overview of the approach

the protein database into high-dimensional feature vectors. These high-dimensional 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). A support vector machine 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 support vector machines, one for each protein class.

During the testing phase, the protein under investigation is first transformed into the high-dimensional feature vector. Each of the trained support vector machines is then queried to determine whether the given protein belongs to the particular protein class in which the support vector machine is trained for. A positive answer will suggest that the protein under investigation has a homologous relationship with the corresponding protein class.

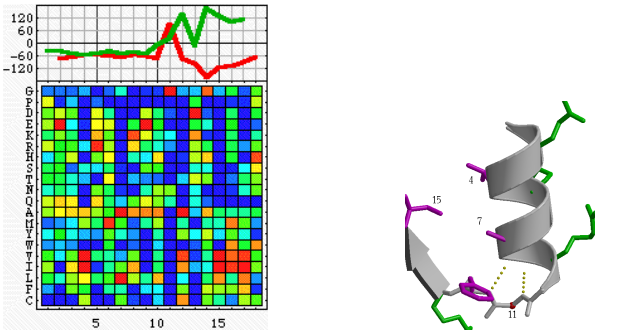
2.1.1 Feature Extraction and Representation

As mentioned earlier, while sequence information does provide important hint to the presence of homologous relationship, it is not sufficient to encompass all the homologous relationships. In fact, there exists a large number of proteins that are homologous but whose sequences are only remotely related. For these remote homology, we observe that their three-dimensional structures share many common characteristics. Thus, capturing these common three-dimensional structures and representing them in a form suitable for the subsequent training and testing of support vector machine algorithms forms one major contribution of our work to the remote homology detection problem.

Since three-dimensional structures are conserved to a greater extent than are their primary structures, the most direct way to combine structure information for homology detection is to encode the three-dimensional structure information into the features. However, three-dimensional protein structures still can not be accurately predicted directly from sequences. An intermediate but useful step is to predict the protein secondary structure, which is a way to simplify the prediction problem by projecting the very complicated 3D 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 [23]. 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 of protein secondary structure prediction includes [6, 10, 26, 18, 9]. Unfortunately, almost all of these methods have not identify the strong relationship between the amino acid sequence and structure. Also, most of these methods focused on three-state secondary structure prediction, namely helix, stand and loop. So, these methods are not appropriate for encoding secondary structure information into feature vectors.

One of the most successful method of predicting local structure is Bystroff and Baker [5]. This is a method for local structure prediction based on a library (I-sites library) of short sequence patterns (profiles) that correlate strongly with protein three-dimensional structure elements. I-sites library is generated through finding correlations between protein sequence and local structure that correlate strongly with protein three-dimensional structural elements. In 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. (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.

To predict the local structure of any unknown protein sequence, sequence patterns (profiles) for each of the 263 clusters of I-sites library were used to score all sub-fragments of this unknown target sequence. Because of the differences in length, the similarity scores of different clusters were not directly comparable; instead it compared the associated "confidence" values. The confidence of a fragment prediction is the probability that a sequence segment with a given score has the predicted structure. Although accuracy is ultimately the most important criterion, the I-sites method has several other apparent advantages: first, secondary structure is predicted throughout a sequence; second, each segment prediction has an associated sequence confidence value that accurately describes the probability that the prediction is

correct; and third, the method is extremely fast, since prediction require only sequence-sequence profile comparisons.

In our problem, given any protein sequence, we use the following method to obtain its structure features: we first segment the given protein sequence into subsequences of length ranging from 7 to 19. For each subsequence, we obtain the probability ("confidence" value) of this subsequence belonging to each of the 263 structure motifs. To minimize effect due to mutation, we apply a threshold such that if the "confidence" value falls below the threshold, it will be set to zero. In addition, it is entirely possible that a protein sequence can occur multiple times. In such cases, a number of heuristics can be used to account for multiple subsequence occurrences. For example, we can take the maximum, the sum, or the average of all the "confidence" value for that protein sequence. We carry out experiments to determine which is the more suitable heuristics and found that using the sum value gives the best performance. Details of the experiment are given in section 5.

By taking the sum value, we can transform a protein sequence into a vector of length 263 where each component denotes the "confidence" value of the presence of the corresponding structure motif. 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.

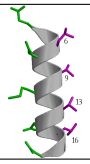
Local Structure	Feature value
	4.76
	3.84
	4.23
...	...

Table 1: A sample of the generated structure feature values

2.1.2 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 [14], neural networks [19], decision tree classifiers [22] etc do not perform well in our remote homology detection problem because of their ineffectiveness in achieving good generalization from relative sparse training data in high dimensions.

It has been established that support vector machines is able to exhibit excellent generalization performance (accuracy on test sets) in practice and have strong theoretical motivation in statistical learning theory [25]. The intuitive idea behind support vector machine is to locate a hyperplane that maximizes the distance separation between the positive and negative examples thereby achieving the best generalization performance. Figure 3 shows the two-dimensional case. Three possible lines are drawn that are able to separate the positive and negative examples. The highlighted line is the one chosen by support vector machine as it maximizes the distance separation between the positive and negative examples.

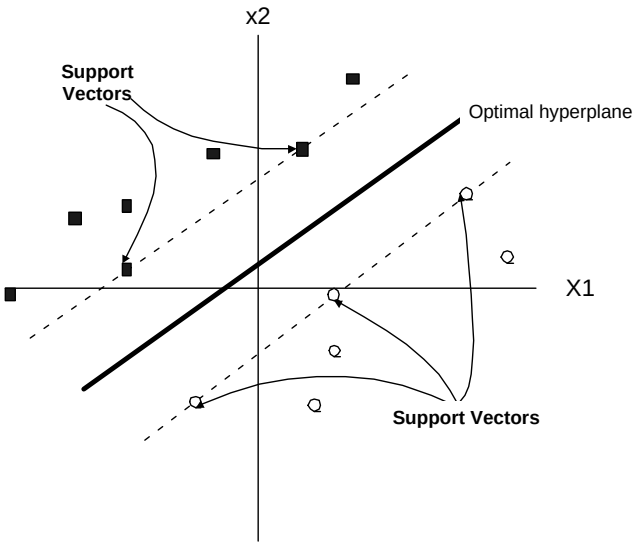


Figure 3: Basic idea of Support Vector Machines

We first train the support vector machines to find such a partitioning hyperplane. Then the support vector machine can predict the classification of an unknown protein by mapping it into the feature space and determine on which side of the hyperplane does the unknown protein lie. Appendix A gives a brief description of the support vector machine methodology.

In our implementation, we use the *gist* support vector machine software implemented by William Stafford Noble and Paul Pavlidis [17]. At the heart of *gist* is 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 only significant parameters needed to tune a SVM are the 'capacity' and the choice of kernel. The capacity allows us to control how much tolerance for errors in the classification of training samples we allow. Capacity therefore affects the generalization ability of the SVM and prevents it from overfitting the training set. We use a capacity equal to 10. This choice of capacity guarantees the numerical stability of the SVM algorithm and provides sufficient generalization.

The second tuning parameter is the kernel. The kernel function allows the SVM to create hyperplanes in high dimensional spaces that effectively separate the training data. Often in the input space training vectors cannot be separated

by a simple hyperplane. The kernel allows transforming the data from one space to another space where a simple hyperplane can effectively separate the data in two classes. We primarily employ the Gaussian kernel for all 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 of the SVM 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 support vector machine trained for that class. The support vector machine classifier produces a 'score' representing the distance of the the testing feature vector from the margin. The larger the score, the further away the vector is from the margin, and the more confident we are of the classifier's prediction.

3. COMPUTATIONAL EFFICIENCY ANALYSIS AND COMPARISON

Computational efficiency is one significant characteristic of any homology detection algorithm. In this respect, SVM-I-sites method is much more efficient than SVM-pairwise and SVM-Fisher method. All of these methods include an SVM optimization and vectorization step. In optimization step, both algorithms take roughly $\mathcal{O}(n^2)$ time, where n is the number of training set examples. The vectorization step of SVM-Fisher requires trainig a profile HMM and computing the gradient vectors. The gradient computation dominates, with a running time of $\mathcal{O}(nmp)$, where m is the length of the longest training set sequence, and p is the number of HMM parameters. The vectorization step of SVM-pairwise involves computing n^2 pairwise scores. Using Smith-Waterman, each computation is $\mathcal{O}(m^2)$, yielding a total running time of $\mathcal{O}(n^2m^2)$. In contrast, SVM-I-sites requires computing the "confidence" value of each sequence containing a pre-defined secondary structure which takes $\mathcal{O}(m)$ time, thus the total running time is $\mathcal{O}(mn)$. Therefore, SVM-I-sites method is mn times fast than SVM-pairwise and p times fast than SVM-Fisher method.

4. EXPERIMENTAL RESULTS

The experiment reported here compare the performance of five algorithms: SVM-I-sites, PSI-BLAST, SAM, SVM-Fisher and SVM-pairwise. We access the recognition performance of each algorithm by testing its ability to classify protein domains into superfamilies in the Structural Classification of Proteins (SCOP)[16] version 1.53. Sequences were selected using the Astral database (astral.stanford.edu [4]), removing similar sequences using an E-value threshold of 10^{-25}. The use of this database allows direct comparison with previous work on remote homology detection method SVM-pairwise. We use the same experiment setup with SVM-pairwise method: for each family, the protein domains within family are considered positive test examples, and the protein domains outside the family but within the same superfamily are taken as positive training examples. The data set yields 54 families containing at least 10 family members (positive train) and 5 superfamily members outside of the family (positive test). 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.

```
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
  if label=1
     tp=tp+1
  else
     fp=fp+1
     roc=roc+tp
  end if
end for

if tp=0
   roc=0
else
   if fp=0
      roc=1
   else
      roc/=tp*fp
   end if
end if
```

Figure 4: 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 5: Algorithm to compute median RFP score

For comparison, we also include the result of PSI-BLAST, SAM and SVM-Fisher methods presented in SVM-pairwise paper. The details of the setup of these methods, please refer to SVM-pairwise paper [15].

Two different scores: receiver operating characteristic (ROC) scores and the median rate of false positives (RFP) are used as the measurements to compare these methods. The ROC score is the area under the receiver operating characteristic curve – the plot of true positives as a function of false positives [8]. 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 algorithm to compute ROC score is shown in Figure 4. The median RFP score is the fraction of negative test sequences that score as high or better than the median-scoring positive sequence. The algorithm to compute median RFP score is shown in Figure 5.

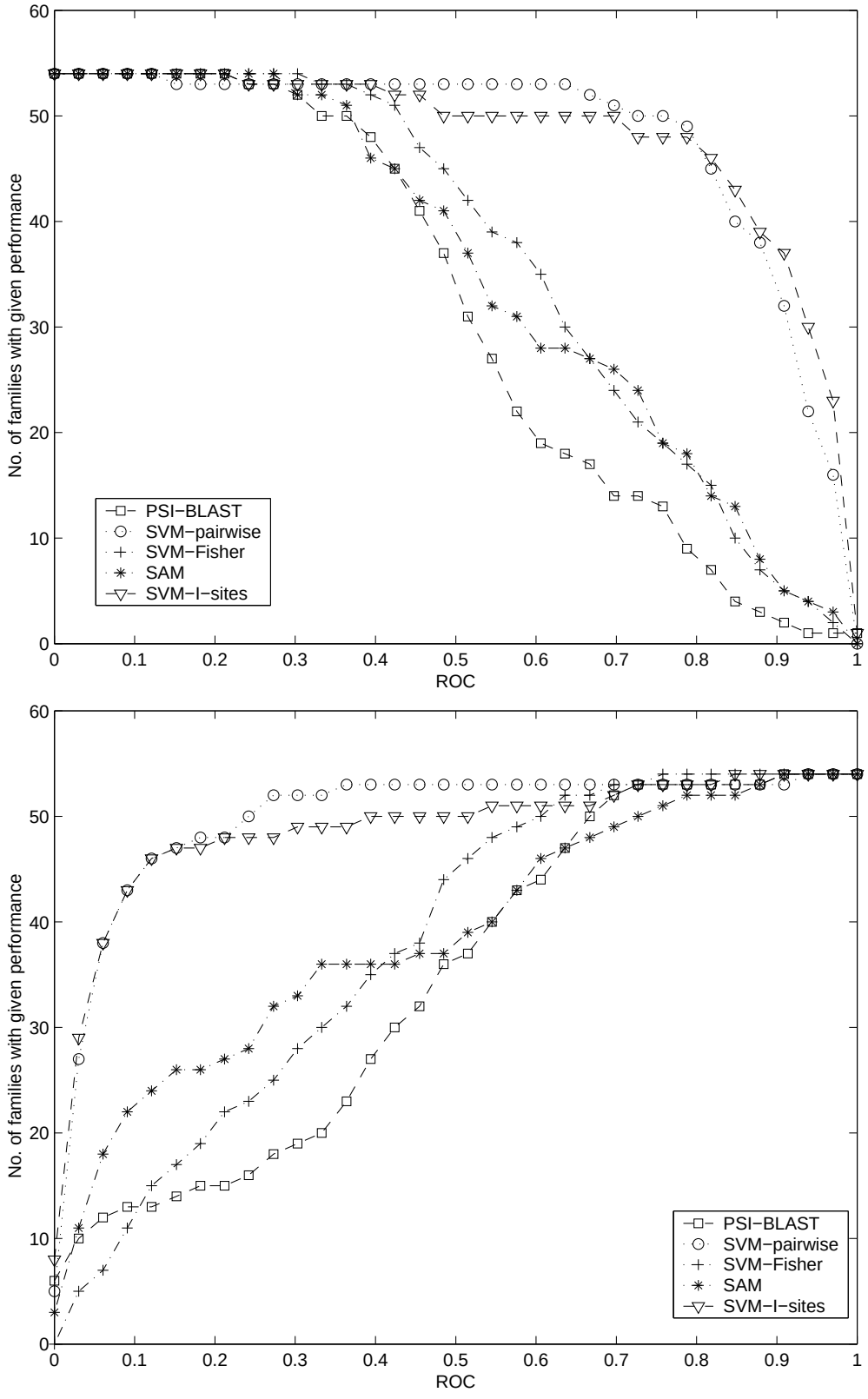


Figure 6: Relative performance of the five homology detection methods. Each graph plots the total number of families for which a given method exceeds a score threshold. The top graph uses ROC scores, and the bottom graph uses median RFP scores. Each series corresponds to one protein homology detection methods.

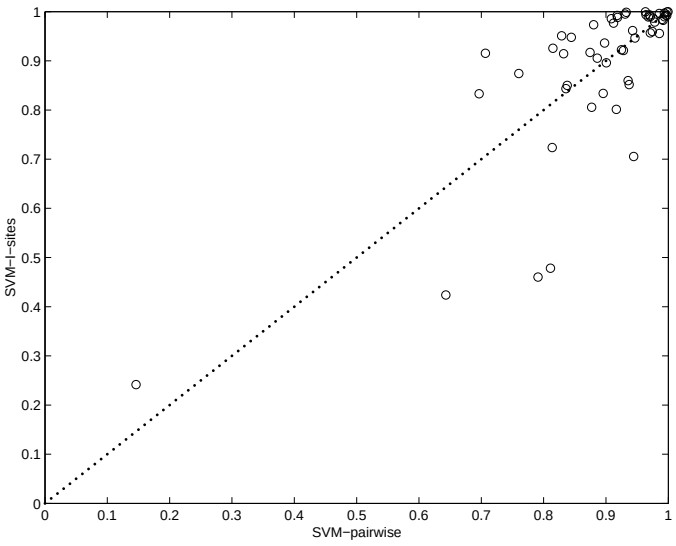


Figure 7: Family-by-family comparison of SVM-pairwise and SVM-I-sites. Each point on the graph corresponds to one of the 54 SCOP superfamilies. The axes are ROC scores achieved by the two primary methods compared in this study: SVM-pairwise and SVM-I-sites

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 2.1.1. 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

The comparison of the five methods are summarized in Figure 6. 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. Using either performance, SVM-I-sites performs significantly better than PSI-BLAST, SAM and SVM-Fisher methods. From the two graphs, we can see that the performance of SVM-I-sites is comparable to SVM-pairwise method. SVM-pairwise adopt pairwise scores as feature values and Smith-Waterman algorithm is recognized the most sensitive pairwise comparison method. While constructing features with local structure probabilities, SVM-I-sites can be an alternative and complimentary method to SVM-pairwise method. This also can be shown from the performance summary Figure 6. Figure 7 is a family-by-family comparison of the 54 ROC scores computed for each method. This figure also suggest SVM-I-sites and SVM-pairwise are two complimentary methods for detection remote homolgy.

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 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. Also, we have developed 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 machine 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 much more efficient in the vectorization step, thus making it a more practical solution for large databases.

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 etc for their helpful discussion and providing experiment result from their prior work.

7. REFERENCES

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

[2] 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 Research*, 25:3389–3402, 1997.

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

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

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

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

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

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

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

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

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

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

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

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

[15] L. Liao and W. S. Noble. Combining pairwise sequence similarity and support vector machines for remote protein homology detection. In *Proceedings of the Sixth International Conference on Computational Molecular Biology*, pages 225–232, 2002.

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

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

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

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

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

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

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

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

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

[25] V. N. Vapnik. *Statistical Learning Theory*. Springer, 1998.

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