A Study of Remote Homology Detection

Beth Logan *Pedro Moreno* *Baris Suzek* *Zhiping Weng* *Simon Kasif*

Cambridge Research Laboratory

Technical Report Series

CRL 2001/05

June 2001

COMPAQ

A Study of Remote Homology Detection

Beth Logan Pedro Moreno
Cambridge Research Laboratory
One Cambridge Center
Cambridge MA 02142

Baris Suzek
Johns Hopkins University
(work on this paper performed during an internship at CRL)

Zhiping Weng Simon Kasif
Bioinformatics Program
Deportment of Bioengineering
Boston University
44 Cummington St
Boston MA 02215

June 2001

Abstract

Functional annotation of newly sequenced genomes is an important challenge for computational biology systems. While much progress has been made towards scaling-up experimental methods for functional assignment to putative genes, most current genomic annotation systems rely on computational solutions for homology modeling via sequence or structural similarity.

We present a new method for remote homology detection that relies on combining probabilistic modeling and supervised learning in high-dimensional features spaces. Our system uses a transformation that converts protein domains to fixed-dimension representative feature vectors, where each feature records the sensitivity of each protein domain to a previously learned set of 'protein motifs' or 'blocks'. Subsequently, the system utilizes Support Vector Machine (SVM) classifiers to learn the boundaries between structural protein classes. Our experiments suggest that this technique performs well relative to several other remote homology methods for the majority of protein domains in SCOP 1.37 PDB90.

Author email: Beth.Logan@compaq.com, Pedro.Moreno@compaq.com

CRL Technical reports are available on the CRL's web page at
http://crl.research.compaq.com.

Compaq Computer Corporation
Cambridge Research Laboratory
One Cambridge Center
Cambridge, Massachusetts 02142 USA

1 Introduction

The last decade has witnessed a consistent surge in sequence information, caused in part by technological breakthroughs in large-scale sequencing and the human genome project. The main challenge facing modern biology is to interpret this newly generated sequence data, and perhaps most significantly, in the short term, to assign function to many putative gene predictions. High-throughput experimental techniques for structural and functional annotations remain relatively elusive, although steady progress is being made. One common solution to functional annotation of putative genes is via structural classification and homology modeling. The traditional and still the most reliable ways to determine the 3-dimensional (3-D) structure of a protein are X-ray crystallography and NMR, which are time-consuming, costly, and currently infeasible for some protein families. Structural genomics initiatives (Burley et al. 1999) have greatly expanded the collection of experimentally determined protein structures. Nevertheless, computational approaches remain, thus far, the only resort for deducing the structure information of many sequences.

Evolutionary pressure forces the retention of sequence features important for structure and function. This has been the main impetus behind homology-based methods, which infer homology from computed sequence similarity. Dynamic programming-based alignment tools such as Smith-Waterman and their efficient approximations such as BLAST (Altshul 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.

Some homologous proteins are sufficiently evolutionarily divergent that they do not exhibit significant sequence similarity. In order to detect such 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. Several protein classification schemes have been developed, namely SCOP (Murzin et al. 1995), CATH (Orango et al. 1999), and FSSP (Holm and Sander 1995). One can build a statistical model for each family or superfamily and then compare a new sequence to a collection of models. Computational methods that relate a sequence to a superfamily-specific model often out-perform pairwise sequence comparison methods.

Examples of superfamily-specific statistical models include sequence profiles (also known as position specific weight matrices) (Altschul et al. 1997) and Hidden Markov Models (HMMs) (Eddy 1998), (Krogh et al 1994), (Delcher et al 1993). These probabilistic models are often called generative because they induce a probability distribution over protein sequences that can subsequently be used to ‘generate’ members of the family using stochastic simulation. Generative models aim to extract features from sequences within a family whose likelihood is high. Generative probabilistic models can be contrasted to discriminative frameworks, which focus on learning the combination of features that discriminate most effectively between families. Support Vector Machine (SVM) and Neural Networks are two popular discriminative methods. A discriminative framework is typically implemented using a classification method or a classifier that learns a boundary between two or more classes.

Most recently a new approach, that combines generative and discriminative methods has been introduced and popularized by (Kasif et al 1998) and (Jaakkola et al

1999). The framework advocated in (Kasif et al 1998) assumes that we first learn a probabilistic model of nature (e.g., the set of all proteins). We then transform every member of the world (e.g., a protein) into a high-dimensional feature vector. The elements of each vector are probabilities or log-likelihood scores that are computed by the probabilistic generative model. Typically, these probabilities are computed with respect to the parameters of the model. After transforming our space to such a feature vector representation, we then use these vectors as input to a learning engine that uses a supervised classification method such as Nearest Neighbor Classifiers (a special case of Kernel Classifiers). It can be shown that this approach can increase the classification accuracy of the generative model when the assumptions made by the model do not accurately conform to reality, which is more common than not.

In a recent independent work by Jaakkola (Jaakkola et al 1999), a generative method (HMM) is combined with a discriminative method (SVM) for detecting remote protein homologies. An HMM is constructed for each protein superfamily. It is then used to compute the gradient of the log-probability of the protein sequence being produced by the HMM with respect to each of the parameters of the HMM (Jaakkola et al 1999). In effect, the protein sequence is transformed into a gradient-log-probability vector. A SVM is then trained on the vectors in order to learn the boundary between each protein superfamily and "the rest of the protein universe". This discriminative model-based approach was shown to be superior to using HMMs alone.

In this paper, we seek to develop an approach that has a natural biological interpretation. Since structural and functional constraints placed on protein families are very complex, our approach aims to facilitate a natural expression of such constraints.

There are two general views that attempt to explain the constraints placed on protein structures: (1) the local view, which considers only 10-20% of residues to be critical, and (2) the global view, which believes that interactions occur along the entire sequence while individual residues contribute minimally. A number of studies found that most single residue mutations do not have measurable effect on protein function and presumably structure (Matthews 1987; Bowie et al. 1990); these support the global view. On the other hand, the local view is supported by (Mirny et al. 1998), who showed that a minimal set of residues is required for the folding of proteins in a physiological timescale. Dosztanyi (Dosztanyi, 1997) reported a minimal set of conserved residues as the stabilization centers in protein structures. It is highly probable that the local view is appropriate for some protein structures and the global view for others. Besides folding constraints, function can dictate conserved sequence and structure features (Kasuya and Thornton 1999).

Conserved motifs tend to appear in linear order in homologous sequences. However exceptions do exist as shown by the DNA methyltransferase family. The catalytic domains of DNA methyltransferases share a common structural fold while having the major functional motifs permuted into three distinct linear orders (Gong et al. 1997). Such protein families pose serious challenges for linear HMMs and their derivatives.

In this paper we present and evaluate an approach similar in spirit to the method developed by Jaakkola et al (1999) but which aims to be simpler and more flexible in terms of representational complexity. We map each protein sequence to an alternate representation in a high-dimensional vector space. We then use SVMs to classify these vectors. The choice of SVMs as a classifier is motivated by its effectiveness in

achieving good generalization from relatively sparse training data in high dimensions (Burges 1998). Our representation of proteins relies on vectors, each component of which corresponds to the similarity of the protein to a structurally or functionally conserved motif, represented by the entries in the BLOCKS database. In other words, we represent a protein domain by recording which of the motifs in BLOCKS it aligns to, and how similar the sequence is to the motif. Our representation coincides with biological intuition, and in principle can be made consistent with local and global folding constraints, functional constraints, as well as sequence non-linearity. The experimental results compare favorably to HMM-based methods for most of the protein families tested in this experiment (see the Discussion section for more details).

2 System and Methods

Our algorithm is implemented in Perl and combines public domain BLOCKS and SVM software. We use a Compaq Tru64 multiprocessor environment.

3 Algorithm

Our procedure for homology detection consists of two major steps. First we convert all the protein sequences of interest to high dimensional feature vectors. We create each vector by scoring a set of pre-learnt motifs against each protein sequence. Once this transformation has taken place, we then learn SVM discriminators to separate each protein family from "the rest of the world". We show this process in Figure 1. The description of each step is given below.

3.1 Feature Vector Generation

The first step of our automated procedure converts each protein sequence or subsequence of interest to a new representation of fixed length. That is, a protein sequence of any length is converted into a feature vector of fixed length. Each dimension of these feature vectors represents the sensitivity of the protein to a particular biological motif. Therefore, in order to create feature vectors, we first create or obtain a database of short, highly conserved regions in related protein domains. Such regions are often called 'blocks', 'motifs' or 'probabilistic templates'.

A motif is represented by a K by L matrix in which the K rows correspond to different amino acids, and L represents the length of the motif. For protein sequences, K = 20. Each cell of the matrix M(amino acid, position) represents the probability or more typically a log-likelihood of seeing that amino acid in that position. Thus, a motif can be thought of as a 0-th order Markov model. A motif of length L is scored against a protein by computing the probability of every subsequence of length L in the protein being generated by the model that corresponds to the motif.

There are a number of databases of short protein motifs available on the Internet, for example EMOTIF at Stanford. The BLOCKS database (Henikoff et al. 1994) is another example. The tool BLIMPS (Wallace and Henikoff 1992) generates a position

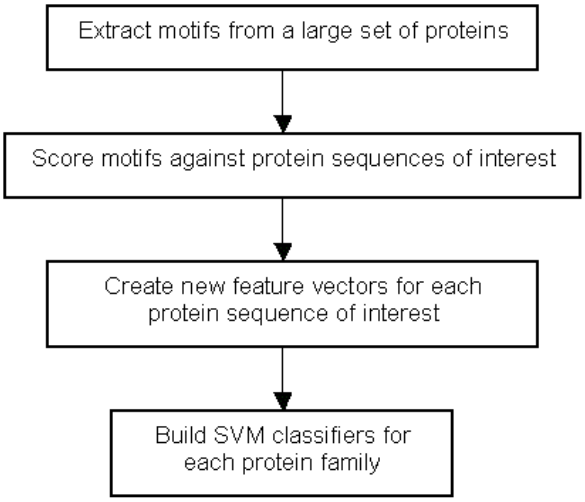

Figure 1: Overall Algorithm

specific scoring matrix for each block in BLOCKS and scores all possible alignments of a protein query sequence using this matrix. For the experiments in this paper, we use BLIMPS to score hits from BLOCKS in order to generate feature vectors for each protein sequence of interest. However, we suspect that by creating a motif database specific to the proteins of interest, even more meaningful feature vectors may be obtained since the motifs from a more general database such as BLOCKS may not occur in the proteins of interest.

To create a feature vector for each protein sequence we search for each motif in the sequence as described above. The result is an N-dimensional feature vector where N is the total number of motifs in our database. In our case the dimensionality is equal to 10000. Each dimension J contains a score describing the degree of alignment of motif J to the protein domain. This process is shown in Figure 2. For the experiments described in this paper (BLOCKS hits on the SCOP 1.37 PDB90 database), this process resulted in very sparse feature vectors (97% sparsity on average).

For the case where a motif is detected multiple times in a domain, we can apply a variety of heuristics. For example, we can take the maximum of all scores for that block in that domain or the sum of such scores. While the sum approach has a better theoretical motivation, in our preliminary experiments, we found that taking the maximum score gives superior classification performance. We can also apply a threshold such that scores below a certain number are set to zero. It is obvious that given the complete set of feature vectors for all protein domains in the training set, we can reduce the dimensionality of these vectors using standard dimension reduction techniques such as Principal Components Analysis (PCA).

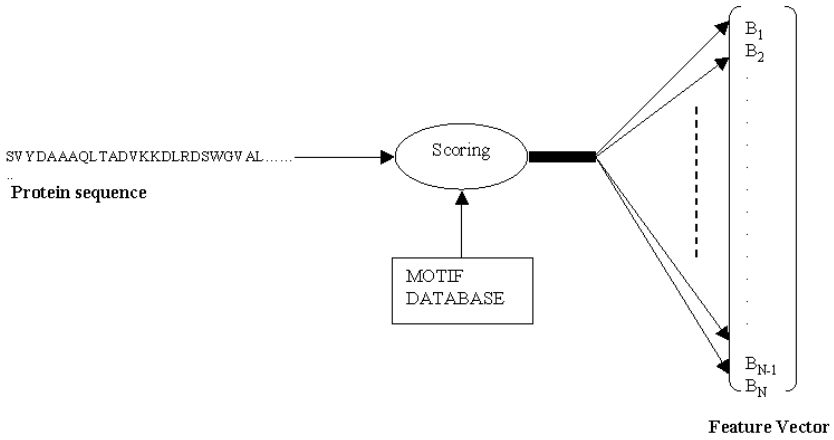

Figure 2: The procedure to generate feature vectors for proteins. Bi in the feature vector is the sum or maximum of the scores (depending on the approach) for the ith block in the motif database found in the protein.

3.2 Construction of SVM Classifiers

Given the labeled feature vectors, we learn Support Vector Machine (SVM) classifiers (Burges 1998) to separate each given structural protein class from "the rest of the world". A SVM classifier learns a separating 'thick' hyperplane between two classes which maximizes the 'margin'. This margin roughly corresponds to the distance between the points residing on the edges of the hyperplane. The appeal of SVMs is twofold. First they do not require any complex tuning of parameters, and second they exhibit a great ability to generalize give a small training corpora. They are particularly amenable for learning in high dimensional spaces. Appendix A gives a short description of the SVM methodology.

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.

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.

In many previously reported applications of SVMs an additional evaluation set is available for tuning the SVM parameters. However, for the experiments reported in this paper no evaluation data is readily available. A commonly found solution deploys

cross-validation. However, this solution is computationally expensive. We use an alternative methodology that aims to avoid the search for the optimal kernel for each classifier by using one that is sufficiently general. We primarily employ the Gaussian kernel for all classifiers. The variance of the associated Gaussian kernel is computed as the median of the distance between all vectors x_1 in Class 1 and x_2 in Class 2. This guarantees that the Gaussian kernel distance is in a reasonable range. In terms of capacity we choose a value that is close to the absolute maximum kernel distance, in our case 10.0. This choice of capacity guarantees the numerical stability of the SVM algorithm and provides sufficient generalization. This solution is a clean way to set the tuned parameters solely based on the training set.

We also report experimental results using a linear support vector machine, which compare favorably with the above solution. This suggest that the main advantage of our method is the choice of representation that combines probabilistic motifs with supervised classification.

An additional tuning step consists of setting the operating point of the classifier to control the amount of false negatives. In our implementation we find a threshold value such that any score returned by the SVM that is bigger than this guarantees no false negatives.

To determine whether an unlabeled protein belongs to a particular structural class, we test it using the SVM created for that class. The SVM classifier produces a 'score' representing the distance of 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 output. If the score is below the threshold set above we classify the vector (and hence the corresponding protein) as belonging to that particular class. Otherwise, it is classified as not belonging to the class.

4 Results

We investigate the performance of our technique on version 1.37 PDB90 of the SCOP database (Murzin et al. 1995). SCOP provides a detailed and comprehensive description of the relationships of all known proteins' structures. The classification is into four hierarchical levels: class, common fold, superfamily and family. Family and superfamily levels describe near and far evolutionary relationships. Fold levels describe geometrical relationships. The unit of classification is the protein domain. In our experiments we investigate how well our method can classify superfamilies.

The use of SCOP 1.37 PDB90 allows direct comparison with previous work on remote homology detection using SVM classifiers in conjunction with vectors generated using HMMs (Jaakola et al. 1999). The training and testing sets used in this previous work are available online from http://www.cse.ucsc.edu/research/compbio/discriminative/ so we are able to duplicate the experiments exactly.

4.1 Testing and Training Sets

SCOP 1.37 PDB90 contains protein domains, no two of which have 90% or more amino acid identity. All SCOP families that contain at least 5 PDB90 sequences and

have at least 10 PDB90 sequences in the other families in their superfamily were selected for positive testing sets, resulting in 33 testing families from 16 superfamilies as listed in Table 1. In the experiments two types of positive training sets are used:

1. A small training set that contains only the PDB90 sequences of all the families of the superfamily containing the positive testing family (except the positive test family).

2. An enhanced (and therefore significantly larger) training set that contains all the homologs found by each individual SAM-T98 (Hughey and Krogh 1998) model built for the selected guide sequences that belong to the superfamily but not to the test itself, in addition to these PDB sequences.

The negative testing and training sets are constructed from PDB sequences in the folds other than the fold containing the positive test family.

4.2 Classification Using HMMER

In addition to comparing our technique of remote homology detection to that described in (Jaakola et al. 1999) we also investigate an HMM-based classifier. This is based on utilities of the HMMER 2 package (Eddy 1998). We describe this technique below.

4.2.1 Model Construction

We build a HMM model for each of the 33 testing families as follows. First we align all the domains in the positive training set without homologs using the multiple alignment tool CLUSTALW (Thompson et al. 1994). Then, using the *hmmbuild* tool, we build HMM models based on these multiple alignments. We use the default arguments of *hmmbuild*.

4.2.2 Model Scoring

We use *hmmsearch* with a very high E-Value to score all proteins in the testing set. These proteins are then ranked based on the Bit- score to determine a threshold that correctly classifies all members of the positive test set (0% false negatives). We then compute the false positive rate produced by this classification and compare to a similarly computed false positive rate computed by the SVM approach.

4.3 Experiments

Table 1 reports the results of our classification experiments. We report the rate of false positives (RFP) at 100% coverage. In other words, we calculate the RFP given 0% false negatives (i.e. zero members of the superfamily misclassified) for each protein family class. This methodology allows us to perform a detailed comparison to prior work on this topic.

Table 1 lists results from four experiments:

Expt.	SCOP Family	SVM HMM HOM	HMMR	SVM MOT	SVM MOT HOM
1	Phycocyanins	0.619	0.471	0.681	0.528
2	Long-chain cytokines	0.123	0.375	0.138	0.092
3	Short-chain cytokines	0.023	0.386	0.021	0.035
4	Interferons/interleukin-10	0.119	0.511	0.012	0.054
5	Parvalbumin	0.000	0.000	0.000	0.000
6	Calmodulin-like	0.000	0.808	0.008	0.000
7	Imm-V Dom	0.016	0.595	0.254	0.006
8	Imm-C1 Dom	0.063	0.738	0.127	0.110
9	Imm-C2 Dom	0.019	0.181	0.303	0.232
10	Imm-I Dom	0.495	0.680	0.164	0.135
11	Imm-E Dom	0.683	0.723	0.852	0.568
12	Plastocyanin/azurin-like	0.772	0.885	0.431	0.753
13	Multidomain & cupredoxins	0.360	0.040	0.705	0.504
14	Plant virus proteins	0.410	0.063	0.501	0.504
15	Animal virus proteins	0.513	0.698	0.770	0.407
16	Legume lectins	0.552	0.312	0.659	0.276
17	Prokaryotic proteases	0.000	0.652	0.031	0.052
18	Eukaryotic proteases	0.000	0.317	0.001	0.000
19	Retroviral protease	0.187	0.394	0.059	0.029
20	Retinoal binding	0.121	0.281	0.344	0.169
21	Alpha-Amylases,N-term	0.037	0.095	0.125	0.086
22	Beta-glycanases	0.079	0.131	0.335	0.440
23	Type II chitinase	0.263	0.145	0.397	0.346
24	Alcohol/glucose dehydro	0.025	0.465	0.008	0.022
25	Glyceraldehyde-3-phosphate	0.107	0.351	0.558	0.224
26	Formate/glycerate	0.004	0.412	0.003	0.024
27	Lactate&malate dehydro	0.074	0.474	0.037	0.019
28	Nucleotide & nucleoside kinases	0.297	0.362	0.000	0.002
29	G proteins	0.051	0.359	0.001	0.001
30	Thioltransferase	0.029	0.540	0.273	0.002
31	Glutathione S-transfer	0.590	0.834	0.871	0.292
32	Fungal lipases	0.007	0.210	0.064	0.014
33	Transferrin	0.072	0.162	0.628	0.389

Table 1: The false positive rates at 100% coverage levels for all 33 test families. See Section 4 for detailed definitions of the experiments.

1. SVM HMM HOM: results reprinted from (Jaakola et al. 1999) for the case with homologs in the training set; SVM HMM HOM stands for SVMs based on HMMs learned over a training set enhanced with homologs.

2. HMMR: remote homology detection using HMMER;

3. SVM MOT: remote homology detection using our technique without homologs in the training set; SVM MOT stands for an SVM method using MOTIFS.

4. SVM MOT HOM: remote homology detection using our technique with homologs in the training set.

The main conclusion reached from the experiments is that the methodology presented here, while conceptually simple achieves higher accuracy than pure HMMs and is at least comparable to previous approaches that combine generative models and supervised classification. Comparative results are summarized in Table 2 and Table 3. Since in some cases our method is superior (or inferior) to the previous methods by less than 5% of the false positive rate which is often not significant, we report the performance of the different methods when differences of less than 2%, 5% and 10% in overall accuracy are judged insignificant. From Table 2 we see that our algorithm is comparable to the SVM HMM technique. From Table 3, we see that our method is typically superior to a pure HMM based approach.

Table 4 documents the performance of our method when we use a relatively simple classification method, basically a using a single hyperplane in 10000 dimensional space. It is clear (basic geometry) that any small set of points in general position (no co-linear points) can be separated in a high-dimensional space (dimension is higher than the number of points) with a linear separator. We therefore used a linear support vector machine to induce this type of a linear decision boundary. Our method essentially ties the previously published results using a simpler representation and classification method.

Definition of Equal	Number of Families		
	SVM HMM HOM superior	SVM MOT HOM superior (our method)	Both techniques 'equal'
Exactly equal	14	16	3
Within 2%	12	13	8
Within 5%	8	11	14
Within 10%	5	7	21

Table 2: Comparison of SVM HMM HOM technique with our proposed SVM MOT HOM technique. This summarizes the number of times each technique is superior for varying definition of 'equal'. Comparison based on the false positive rate at 100% coverage from Table 1

In order to provide insight on the distribution of protein motifs among the different families we computed a histogram that documents the number of 'hits' per protein.

Definition of Equal	Number of Families		
	HMMR superior	SVM MOT superior (our method)	Both techniques 'equal'
Exactly equal	14	18	1
Within 2%	14	18	1
Within 5%	12	18	3
Within 10%	10	18	5

Table 3: Comparison of HMMR with our proposed SVM MOT technique. This summarizes the number of times each technique is superior for varying definition of 'equal'. Comparison based on the false positive rate at 100% coverage from Table 1

Definition of Equal	Number of Families		
	SVM HMM HOM superior	SVM MOT HOM superior (our method)	Both techniques 'equal'
Exactly equal	15	15	3
Within 2%	12	13	8
Within 5%	10	10	13
Within 10%	5	5	23

Table 4: Comparison of SVM HMM HOM with our proposed SVM MOT HOM technique using linear support vector machines, a particularly simple classification scheme. This summarizes the number of times each technique is superior for varying definition of 'equal'. Comparison based on the false positive rate at 100% coverage.

This distribution is depicted in Figure 3. It appears to be a Normal Distribution with a mean around 300+ hits.

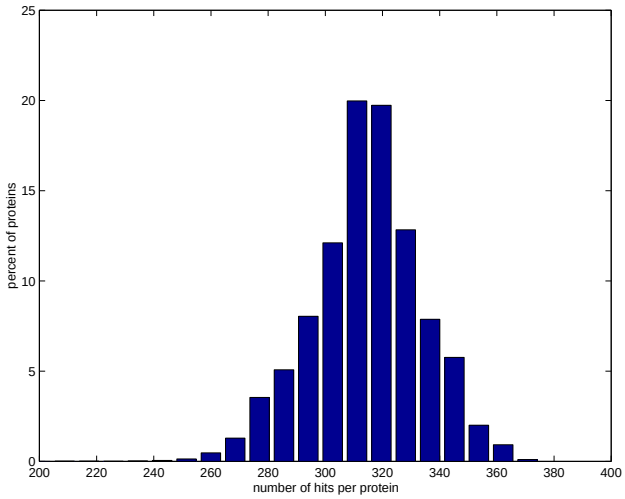

Figure 3: Histogram of number of BLOCKS hits for proteins from SCOP.

We also plot the probability distribution for the number of proteins hit by a single motif. The only 'surprising' finding to report here is that a small number of motifs hit almost all of the proteins in the database. This could be simply a result of a low complexity region, or a previously missed phenomenon that needs to be studied further. This distribution is depicted in Figure 4.

5 Discussion

The main novelty of our technique is our method of constructing feature vectors and the combination of this representation with a classification method capable of learning in very sparse high-dimensional spaces. Each component of our protein-vectors represents the sensitivity of the protein domain to a given 'motif'. At present, we use generic blocks from the BLOCKS database (Henikoff et al. 1994). However, the feature vectors generated for SCOP PDB90 using this technique are very sparse since many BLOCKS are not found in many domains. We believe even better results could be achieved by constructing a SCOP-specific database of motifs.

The motivation for this approach has both biological and statistical underpinnings. In terms of biology, short sequence motifs that 'cover' the space of all proteins provide a promising starting point for analyzing a particular sequence. Statistically, if a particular short motif occurs in a large number of families it is better to learn it across the entire set of proteins. This approach is similar to the learning of phonemes in speech processing where the dictionary items are often learned across the entire corpora and

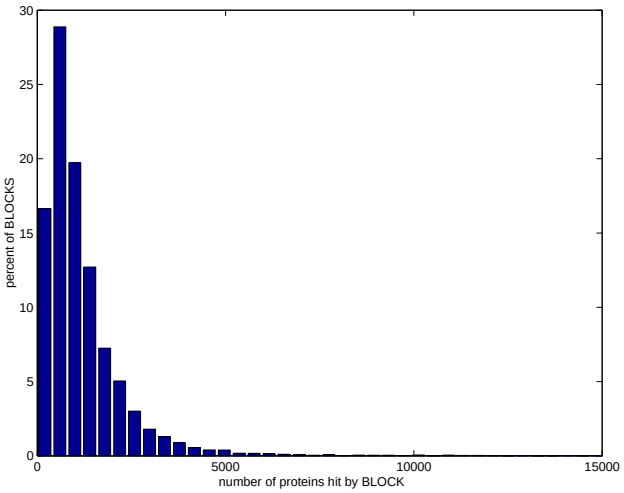

Figure 4: Probability distribution for the number of proteins hit by a BLOCKS motif. Computed using proteins from SCOP.

then words are learned as combination of the dictionary terms (e.g. see (Rabiner and Juang 1993)).

An HMM based approach learns only the parameters that provide a fit of a particular model architecture to a given protein family. Using linear profile HMMs it is difficult to describe certain relationships between short protein motifs. Other HMM architectures are possible but these require developing different models for different families. Learning the structure of HMMs automatically from data is typically difficult and requires large training corpora. The combined generative-supervised learning methodology proposed here allows us substantial flexibility in learning the particular composition of sequence motifs which can include order and other learned or prespecified constraints.

Previous classification of protein domains based on motifs (e.g. blocks) typically rely on simple classification rules. For example, a rule might be: if at least 5 specific motifs occur in a sequence then classify the sequence as Kinase domain. Our SVM approach generalizes these simple rules and provides a systematic way to learn classification rules combining all known motifs. The ability of SVMs to learn in sparsely sampled high- dimensional spaces is the key to producing effective results based on this methodology.

This approach is in part justified by the recent success of David Baker and colleagues' methodology aimed at ab initio protein structure prediction (Simons et al. 1997; Simons et al. 1999; Simons et al. 1999). The Baker approach builds the 3-D structure of a protein by assembling its 9-residue fragments with local sequence similarity to any protein of known structure. The interactions between these fragments are scored explicitly using features such as the burial of hydrophobic residues and the

assembly of beta-strands into beta-sheets. Here we also represent a protein sequence by an array of short sequence fragments - the BLOCKS motifs. Instead of scoring the interactions between the motifs, we use SVM to 'learn' the characteristic interactions in a protein family and to discriminate between different families.

6 Acknowledgments

The authors thank Tommi Jaakkola who provided many valuable comments on this paper as well as suggesting the specific methodology for automatically setting the variance of the SVM classifier.

7 References

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., et al. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, 25(17), 3389-402.

Bowie, J. U., Reidhaar-Olson, J. F., Lim, W. A. and Sauer, R. T. (1990). Deciphering the message in protein sequences: tolerance to amino acid substitutions. *Science*, 247(4948), 1306-10.

Burley, S. K., Almo, S. C., Bonanno, J. B., Capel, M., Chance, M. R., et al. (1999) Structural genomics: beyond the human genome project. *Nat Genet* 23(2), 151-7.

Burges, C. (1998) A tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery Journal*.

Delcher, A. L., Kasif, S., Goldberg, H. R. and Hsu, B. (1993) Protein Secondary-Structure Modeling with Probabilistic Networks. In Proceedings *International Conference on Intelligent Systems and Molecular Biology*, pp. 109–117.

Dosztanyi, Z., Fiser, A. and Simon, I. (1997) Stabilization centers in proteins: identification, characterization and predictions. *J Mol Biol* 272(4), 597-612.

Eddy, S. (1998) http://hmmer.wustl.edu.

Gong, W., O'Gara, M., Blumenthal, R. M. and Cheng, X. (1997) Structure of pvu II DNA- (cytosine N4) methyltransferase, an example of domain permutation and protein fold assignment. *Nucleic Acids Research* 25(14), 2702-15.

Henikoff, S. S. and Henikoff, J. G. (1994) Protein family classification based on searching a database of blocks. In *Genomics* 19, 97–107.

Hughey, R. and Krogh, A. (1998) *Sequence Alignment and Modeling Software System*, Technical Report UCSC-CRL-96-22.
http://www.cse.ucsc.edu/research/compbio/sam.html.

Holm, L. and Sander, C. (1995) Searching Protein Structure Databases has come of age. In *Proteins* 19, 165–173.

Jaakkola, T., Diekhans, M. and Haussler, D. (1999) A discriminative framework for detecting remote protein homologies, In Proceedings *Seventh International Conference on Intelligent Systems for Molecular Biology*, pp. 149–158.

Kasif, S., Salzberg, S., Waltz, D., Rachlin, J. and Aha, D. (1998) Towards a Probabilistic Framework for Memory-Based Reasoning. In *Artificial Intelligence*, pp.

287–311, 1998.

Kasuya, A. and Thornton, J. M. (1999) Three- dimensional structure analysis of PROSITE patterns. *J Mol Biol* 286(5), 1673-91.

Krogh, K., Brown, M., Mian, I. S., Sjolander, K. and Haussler, D. (1994) Hidden Markov models in computational biology: Applications to protein modeling. *Journal of Molecular Biology* 235, 1501–1531.

Matthews, B. W. (1987) Genetic and structural analysis of the protein stability problem. *Biochemistry* 26(22), 6885-8.

Mirny, L. A., Abkevich, V. I. and Shakhnovich, E. I. (1998) How evolution makes proteins fold quickly. *Proc Natl Acad Sci U S A* 95(9), 4976-81.

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.

Orengo, C. A., Pearl, F. M., Bray, J. E., Todd, A. E., Martin, A. C., Lo Conte, L., Thornton, J. M. (1999) The CATH Database provides insights into protein structure/function relationships. *Nucleic Acids Res.* 27(1), 275–279.

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

Rabiner, L. R. and Juang, B-H. (1993) *Fundamentals of Speech Recognition.* Prentice-Hall.

Simons, K. T., Bonneau, R., Ruczinski, I. and Baker, D. (1999) Ab initio protein structure prediction of CASP III targets using ROSETTA. *Proteins Suppl* (3), 171-6.

Simons, K. T., Kooperberg, C., Huang, E. and Baker, D. (1997) Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *J Mol Biol* 268(1), 209-25.

Simons, K. T., Ruczinski, I., Kooperberg, C., Fox, B. A., Bystroff, C., et al. (1999) Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins. *Proteins* 34(1), 82-95.

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

Wallace, J. C. and Henikoff, S. (1992) PATMAT: a searching and extraction program for sequence, pattern, and block queries and databases. *CABIOS* 8, 249–254.

8 Appendix A - A Short Description of Support Vector Machines

Given a set of training samples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$ with labels $y_1, y_2, \ldots, y_m$ we aim to learn the hyperplane $\mathbf{w}.\mathbf{x} + b$ which separates the data into classes such that:

$$\begin{aligned} \mathbf{w}.\mathbf{x}_i + b &\geq 1 - \xi_i \text{ if } y_i = 1 \\ \mathbf{w}.\mathbf{x}_i + b &\leq \xi_i - 1 \text{ if } y_i = -1 \\ \xi_i &\geq 0 \ \forall i. \end{aligned}$$

For no misclassifications ($\xi_i = 0$), we find the separating hyperplane which maximizes the distance between it and the closest training sample. It can be shown that this is equivalent to maximizing $2/|\mathbf{w}|$ subject to the constraints above. By forming the Lagrangian and solving the dual problem, this can be translated into the following:

$$\begin{aligned} \text{minimize}: &\quad \textstyle\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\ \text{subject to}: &\quad \alpha_i \geq 0 \text{ and } \textstyle\sum_i \alpha_i y_i = 0. \end{aligned}$$

where α_i is a Lagrange multiplier. There is one Lagrange multiplier for each training sample. The training samples for which the Lagrange multipliers are non-zero are called *support vectors*. Samples for which the corresponding Lagrange multiplier is zero can be removed from the training set without affecting the position of the final hyperplane. The above formulation is a well understood quadratic programming problem for which solutions exist. The solution may be non- trivial though in cases where the training set is large.

If no hyperplane exists (because the data is not linearly separable) we add penalty terms ξ_i to account for misclassifications. We then minimize $|\mathbf{w}|^2/2 + C \sum_i \xi_i$ where C, the *capacity*, is a parameter which allows us to specify how strictly we want the classifier to fit the training data. This can be translated to the following dual problem:

$$\begin{aligned} \text{minimize}: &\quad \textstyle\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\ \text{subject to}: &\quad 0 \leq \alpha_i \leq C \text{ and } \textstyle\sum_i \alpha_i y_i = 0. \end{aligned}$$

which again can be solved by standard techniques.

The classification framework outlined above is limited to linear separating hyperplanes. It is possible however to use a non-linear hyperplane by first mapping the sample points into a higher dimensional space using a non-linear mapping. That is, we choose a map $\phi : \Re^n \rightarrow \Im$ where the dimension of $\Im$ is greater than n . We then seek a separating hyperplane in the higher dimensional space. This is equivalent to a non-linear separating surface in $\Re^n$.

As shown above, the data only ever appears in our training problem in the form of dot products, so in the higher dimensional space the data appears in the form $\phi(\mathbf{x}_i).\phi(\mathbf{x}_j)$.

If the dimensionality of $\Im$ is very large, this product could be difficult or expensive to compute. However, by introducing a *kernel* function such that $K(\mathbf{x}_i.\mathbf{x}_j) = \phi(\mathbf{x}_i).\phi(\mathbf{x}_j)$ we can use this in place of $\mathbf{x}_i.\mathbf{x}_j$ everywhere in the optimization problem and never need to know explicitly what ϕ is. Some examples of kernel functions are the polynomial kernel $K(\mathbf{x}_i.\mathbf{x}_j) = (\mathbf{x}_i.\mathbf{x}_j + 1)^p$ and the Gaussian radial basis function (RBF) kernel $K(\mathbf{x}_i.\mathbf{x}_j) = e^{|\mathbf{x}_i - \mathbf{x}_j|^2/2\sigma^2}$.

After solving for $\mathbf{w}$ and b we determine which class a test vector $\mathbf{x}_i$ belongs to by evaluating $\mathbf{w}.\mathbf{x}_i + b$ or $\mathbf{w}.\phi(\mathbf{x}_i) + b$ if a transform to a higher dimensional space has been used. It can be shown that the solution for $\mathbf{w}$ is given by $\bar{w} = \sum_i \alpha_i y_i \mathbf{x}_i$. Therefore, $\mathbf{w}.\phi(\mathbf{x}_t)$ can be rewritten

$$\begin{aligned} \mathbf{w}.\phi(\mathbf{x}_t) + b &\equiv \sum_i \alpha_i y_i \phi(\mathbf{x}_i).\phi(\mathbf{x}_t) + b \\ &\equiv \sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_t) + b. \end{aligned}$$

Thus we again can use the Kernel function rather than actually making the transformation to higher dimensional space since the data appears only in dot product form.

CRL 2001/05 A Study of Remote Homology Detection

June 2001

Beth Logan Pedro Moreno Baris Suzek Zhiping Weng Simon Kasif

COMPAQ