Using the Fisher kernel method to detect remote protein homologies

Tommi Jaakkola*, **Mark Diekhans**, **David Haussler**
Department of Computer Science
University of California
Santa Cruz, CA 95064
tommi@ai.mit.edu, markd@cse.ucsc.edu, haussler@cse.ucsc.edu

Abstract

A new method, called the Fisher kernel method, for detecting remote protein homologies is introduced and shown to perform well in classifying protein domains by SCOP superfamily. The method is a variant of support vector machines using a new kernel function. The kernel function is derived from a hidden Markov model. The general approach of combining generative models like HMMs with discriminative methods such as support vector machines may have applications in other areas of biosequence analysis as well.

Introduction

Many statistical, sequence-based tools have been developed for detecting protein homologies. These include BLAST (Altshul *et al.* 1990; Altschul *et al.* 1997), Fasta (Pearson & Lipman 1988), PROBE (Neuwald *et al.* 1997), templates (Taylor 1986), profiles (Gribskov, McLachlan, & Eisenberg 1987), position-specific weight matrices (Henikoff & Henikoff 1994), and Hidden Markov Models (HMMs) (Krogh *et al.* 1994). Recent experiments (Brenner 1996; Park *et al.* 1998) have used the SCOP classification of protein structures (Hubbard *et al.* 1997) to test many of these methods to see how well they detect remote protein homologies that exist between protein domains that are in the same structural superfamily, but not necessarily in the same family. This work has shown that methods such as PSI-BLAST and HMMs, which build a statistical model from multiple sequences, perform better than simple pairwise comparison methods, but all sequence-based methods miss many important remote homologies.

We present and evaluate a new methodology for detecting remote protein homologies. In this approach we use generative statistical models built from multiple sequences, in this case HMMs, as a way of extracting features from protein sequences. This maps all protein sequences to points in a Euclidean feature space of fixed dimension. (See (Linial *et al.* 1997) for a different method of mapping protein sequences into Euclidean space). We then use a general discriminative statistical method to classify the points representing protein sequences by domain superfamily. This is quite distinct from methods that train the parameters of the HMM itself to give a more discriminative model (Eddy, Mitchison, & Durbin 1995; Mamitsuka 1996). Other discriminative methods, using neural nets, are described in (Dubchak *et al.* 1995; Dubchak, Muchnik, & Kim 1997). Using our method, we obtain a substantial improvement in identifying remote homologies over what is achieved by HMMs alone, as they are currently employed. This new method also compares favorably to what have been called family pairwise search homology methods, in which the scores from all pairwise comparisons between a query protein and the members of a known protein family are combined to improve performance (Grundy 1998).

*Current address: MIT AI Lab, 545 Technology Square, Cambridge MA 02139.

Methods

The statistical modeling approach to protein sequence analysis involves constructing a *generative* probability model, such as an HMM, for a protein family or superfamily (Durbin *et al.* 1998). Sequences known to be members of the protein family are used as *(positive) training examples.* The parameters of a statistical model representing the family are estimated using these training examples, in conjunction with general *a priori* information about properties of proteins. The model assigns a probability to any given protein sequence. If it is a good model for the family it is trained on, then sequences from that family, including sequences that were not used as training examples, yield a higher probability score than those outside the family. The probability score can thus be interpreted as a measure of the extent to which a new protein sequence is homologous to the protein family of interest. Considerable recent work has been done in refining HMMs for the purpose of identifying weak protein homologies in this way (Krogh *et al.* 1994; Baldi *et al.* 1994; Eddy 1995; Hughey & Krogh 1996; Karplus *et al.* 1997).

Let $X = [x_1, ..., x_n]$ denote a protein sequence, where each x_i is an amino acid residue. Suppose that we are interested in a particular protein family such as immunoglobulins and have estimated an HMM H_1 for this family (for details of the estimation process see, e.g., (Durbin *et al.* 1998)). We use $P(X|H_1)$ to denote the probability of X under the HMM, and $P(X|H_0)$ to denote the probability of X under the null model. The score used in database search is the likelihood ratio

$$\begin{aligned}\mathcal{L}(X) &= \log \frac{P(X|H_1)P(H_1)}{P(X|H_0)P(H_0)} \qquad (1)\\ &= \log \frac{P(X|H_1)}{P(X|H_0)} + \log \frac{P(H_1)}{P(H_0)}\end{aligned}$$

A positive value of the likelihood ratio $\mathcal{L}(X)$ is taken as an indication that the new sequence X is indeed a member of the family. The constant factor $\log P(H_1)/P(H_0)$, the log prior odds, provides an *a priori* means for biasing the decision and does not affect the ranking of sequences being scored.

Discriminative approaches

The parameters of a generative model are estimated in such a way as to make the positive training examples, proteins in the family being modeled, very likely under the probability model. In contrast, the parameters of a discriminative model are estimated using both positive training examples and *negative training examples*, which are proteins that are not members of the family being modeled. The goal in estimating the parameters of a discriminative model is to find parameters such that the score derived from the model can be used to discriminate members of the family from non-members, e.g. such that members of the family receive a high score and non-members receive a low score.

Although the likelihood ratio above is optimal when the HMM and the null model are perfectly accurate models of the data, it can perform poorly when these models are not accurate. This can easily happen with limited training sets (Barrett, Hughey, & Karplus 1997; Park *et al.* 1998). Discriminative methods can be used to directly optimize the decision rule using both negative and positive examples, and often perform better.

Kernel methods

Suppose we have a training set of examples (protein sequences) $\{X_i\}, i = 1, \ldots, n$ for which we know the correct hypothesis class, H_1 or H_0. In other words, we have a set of protein sequences that are known to be either homologous to the family of interest or not. We model the discriminant function $\mathcal{L}(X)$ directly via the following expansion in terms of the training examples:

$$\begin{aligned}\mathcal{L}(X) &= \log P(H_1|X) - \log P(H_0|X) \qquad (2)\\ &= \sum_{i:X_i \in H_1} \lambda_i K(X, X_i) - \sum_{i:X_i \in H_0} \lambda_i K(X, X_i)\end{aligned}$$

The sign of the discriminant function determines the assignment of the sequences into hypothesis classes. The contribution, either positive or negative, of each training example (sequence) to the decision rule consists of two parts: 1) the overall importance of the example X_i as summarized with the non-negative coefficient λ_i and 2) a measure of pairwise "similarity" between the training example X_i and the new example X, expressed in terms of a *kernel* function $K(X_i, X)$. The user supplies the kernel function appropriate for the application area. This is the most critical component. The free parameters in the above decision rule are the coefficients λ_i; these can be estimated by iteratively maximizing a quadratic objective function as described in (Jaakkola, Diekhans, & Haussler 1998b).

The Fisher kernel

Finding an appropriate kernel function for a particular application area can be difficult and remains largely an unresolved issue. We have, however, developed a general formalism for deriving kernel functions from generative probability models (Jaakkola & Haussler 1998). This formalism carries several advantages, including the ability to handle complex objects such as variable length protein sequences within the kernel function. Furthermore, the formalism facilitates the encoding of prior knowledge about protein sequences, via the probability models, into the kernel function.

Our approach here is to derive the kernel function from HMMs corresponding to the protein family of interest. We are thus able to build on the work of others towards adapting HMMs for protein homology detection(Krogh *et al.* 1994; Hughey & Krogh 1996; Karplus *et al.* 1997). Our use of protein models in the kernel function, however, deviates from the standard use of such models in biosequence analysis in that the kernel function specifies a similarity score for any pair of sequences, whereas the likelihood score from an HMM only measures the closeness of the sequence to the model itself.

We extract and entertain a richer representation for each sequence in the form of what are known as *sufficient statistics*. In the context of HMMs these statistics are computed with each application of the standard forward-backward algorithm (Rabiner & Juang 1986). The fixed length vector of sufficient statistics contains a value for each parameter in the HMM and these values are the posterior frequencies of having taken a particular transition or having generated one of the residues of the query sequence X from a particular state. The vector of sufficient statistics thus reflects the process of generating the query sequence from the HMM. More generally, they provide a complete summary of the sequence in the parameter space of the model.

The idea of using sufficient statistics as an intermediate representation of the query sequence can be generalized considerably(Jaakkola & Haussler 1998). In the more general treatment, one works not with the vector of sufficient statistics directly but with an analogous quantity known as the *Fisher score*

$$U_X = \nabla_\theta \log P(X|H_1, \theta) \tag{3}$$

Each component of U_X is a derivative of the log-likelihood score for the query sequence X with respect to a particular parameter. The magnitude of the components specify the extent to which each parameter contributes to generating the query sequence. The computation of these gradients in the context of HMMs along with their relation to sufficient statistics is described in more detail in (Jaakkola, Diekhans, & Haussler 1998b).

Finding an appropriate kernel function in this new gradient representation is easier than in the original space of variable length protein sequences.We only need to quantify the similarity between two fixed length gradient vectors U_X and $U_{X'}$ corresponding to two sequences X and X', respectively. The kernel function used in our experiments is given by

$$K(X, X') = e^{-\frac{1}{2\sigma^2}(U_X - U_{X'})^T (U_X - U_{X'})} \tag{4}$$

as derived in (Jaakkola, Diekhans, & Haussler 1998b). The scaling parameter σ appearing in this kernel was set equal to the median Euclidean distance between the gradient vectors corresponding to the training sequences in the protein family of interest and the closest gradient vector from a protein belonging to another protein fold.

To summarize, we begin with an HMM trained from positive examples to model a given protein family. We use this HMM to map each new protein sequence X we want to classify into a fixed length vector, its Fisher score, and compute the kernel function on the basis of the Euclidean distance between the score vector for X and the score vectors for known positive and negative examples X_i of the protein family. The resulting discriminant function is given by

$$\mathcal{L}(X) = \sum_{i:X_i \in H_1} \lambda_i K(X, X_i) - \sum_{i:X_i \in H_0} \lambda_i K(X, X_i), \tag{5}$$

where K is the kernel function defined above and the λ_i are estimated from the positive and negative training examples X_i. We refer to this as the *SVM-Fisher* method.

Combination of scores

In many cases we can construct more than one HMM model for the family or superfamily of interest. It is advantageous in such cases to combine the scores from the multiple models rather than selecting just one. Let $\mathcal{L}_i(X)$ denote the score for the query sequence X based on the i^{th} model. This might be the score derived from the SVM-Fisher method, Equation (5), or the log likelihood ratio for the generative HMM model, $\log \frac{P(X|H_1)}{P(X|H_0)}$, or even a negative log E-value derived from a BLAST comparison, as described in he methods section. We would like to combine the SVM-Fisher scores for X from all the models of the given family, and similarly with HMM and BLAST scores. Since there is no clearly optimal way to combine these scores in practice, we explore here only two simple heuristic means. These are *average score*

$$\mathcal{L}_{ave}(X) = \frac{1}{N} \sum_i \mathcal{L}_i(X) \tag{6}$$

and *maximum score*

$$\mathcal{L}_{max}(X) = \max_i \mathcal{L}_i(X), \tag{7}$$

where in each case the index i ranges over all models for a single family of interest. These combination methods have also been explored in other protein homology experiments (Grundy 1998). The average score method works best if the scores for the individual models are fairly consistent, and the maximum score method is more appropriate when we expect larger values of some individual model scores to be more reliable indicators. We have found that the maximum score method works better in our experiments with generative HMM models and BLAST scores, so this approach is used there. However, the average score method works better for combining scores from our discriminative models, so it is used in these experiments.

Experimental Methods

We designed a set of experiments to determine the ability of SVM-Fisher kernel discriminative models to recognize remote protein homologs. The SVM-Fisher kernel methods were compared to *BLAST* (Altshul *et al.* 1990; Gish & States 1993) and the generative HMMs built using the *SAM-T98* methodology (Park *et al.* 1998; Karplus, Barrett, & Hughey 1998; Hughey & Krogh 1995; 1996). The experiments measured the recognition rate for members of superfamilies of the SCOP protein structure classification (Hubbard *et al.* 1997). We simulate the remote homology detection problem by withholding all members of a SCOP family from the training set and training with the remaining members of the SCOP superfamily. We then test sequences from the withheld family to see if they are recognized by the model built from the training sequences. Since the withheld sequences are known remote homologs, we are able to demonstrate the relative effectiveness of the techniques in classifying new sequences as remote members of a superfamily. In a sense, we are asking, "Could the method discover a new family of a known superfamily?"

Overview of experiments

The SCOP version 1.37 PDB90 domain database, consisting of protein domains, no two of which have 90% or more residue identity, was used as the source of both training and test sequences. PDB90 eliminates a large number of essentially redundant sequences from the SCOP database. The use of the domain database allows for accurate determination of a sequence's class, eliminating the ambiguity associated with searching whole-chain protein databases.

The generative models were obtained from an existing library of *SAM-T98* HMMs. The *SAM-T98* algorithm, described more fully in (Karplus, Barrett, & Hughey 1998), builds an HMM for a SCOP domain sequence by searching the non-redundant protein database *NRP* for a set of potential homologs of the sequence and then iteratively selecting positive training sequences from among these potential homologs and refining a model. The resulting model is stored as an alignment of the domain sequence and final set of homologs.

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 our test, resulting in 33 test families from 16 superfamilies (Jaakkola, Diekhans, & Haussler 1998b). When testing the recognition of one these families, the training and test sets were constructed as follows. The positive training examples were selected from the remaining families in the superfamily containing the family in question and the negative training examples from outside of the fold that the family belongs to. The positive test examples consisted of all the PDB90 sequences in the family. The negative test examples were chosen from outside of the fold containing the family and in such a way that the negative examples in the training set and those in the test set never came from the same fold. Figure 1 shows an example of the division.

For each of the 33 test families, all the test examples, both positive and negative, were scored, based on a discriminant function obtained from the training examples for that family. We used various methods, described in the following section, to measure how well the discriminant function performed by assessing to what extent it gave better scores to the positive test examples than it gave to the negative test examples. Using this setup, the performance of the SVM-kernel method was compared to the performance of the generative HMM alone, and to BLAST scoring methods.

Multiple models used

After selecting a test family, we must construct a model for its superfamily using available sequences from the other families in that superfamily. The *SAM-T98* method starts with a single sequence (the guide sequence for the domain) and builds a model. In general, there are too many sequences in the other families of the superfamily to consider building a model around each one of them. So we used a subset of PDB90 superfamily sequences present in a diverse library of existing HMMs. The SVM-Fisher method was subsequently trained using each of these models in turn. The scores for the test sequences, given each HMM model, were computed from Equation (5), and the scores obtained based on multiple models were combined according to Equation (6).

Details on the training and test sets

In each experiment, all PDB90 sequences outside the fold of the test family were used as either negative training or negative test examples. All experiments were repeated with the test/training allocation of negative examples reversed. This resulted in approximately 2400 negative test sequences for most test families. The split of negative examples into test and training was done on a fold-by-fold basis, in such a way that folds were never split between test and train. This insured that a negative training example was never similar to a negative test example, which might give a significant advantage to discriminative methods. In actual applications, this requirement could be relaxed, and further improvements might be realized by using discriminative methods.

For positive training examples, in addition to the PDB90 sequences in the superfamily of the test family (but not in the test family itself), we used all of the homologs found by each individual *SAM-T98* model built for the training PDB90 sequences.

BLAST methods

Two *BLAST* methodologies were used for comparison, each using WU-BLAST version 2.0a16 (WU-BLAST ; Althschul & Gish 1996). These are family pairwise search homology methods, as explored in (Grundy 1998). In both methods, the PDB90 database was queried with each positive training sequence, and E-values were recorded. One method, referred to as *BLAST:SCOP-only* in the results section, used positive training examples as defined by (1) above. The other, which we call *BLAST:SCOP+SAM-T98-homologs*, included the *SAM-T98* domain homologs as positive examples, as in (2) above. In both cases, the scores were combined by the maximum method, so the final score of a test sequence in the PDB90 database was taken to be the maximum $-\log$ E-value for any of the positive training example query sequences. This score measures the BLAST-detectable similarity of the test sequence to the closest sequence in the set of positive training sequences. In (Grundy 1998), a related combination rule, which instead used the average of the BLAST bit scores, was suggested. We tried a similar average method, taking the average of the $-\log$ E-values, which should in theory be more accurate than averaging the bit scores. However, the maximum method performed best, so we report results for that combination method only.

Generative HMM scores

Finally, we also report results using the *SAM-T98* method as a purely generative model. The null model used here is the reverse sequence model from (Park *et al.* 1998; Karplus, Barrett, & Hughey 1998). We used the same data and the same set of models as in the SVM-Fisher score experiments; we just replaced the SVM-Fisher score with the *SAM-T98* score. However, the scores were combined with the maximum method, since that performed slightly better in this case.

In these experiments we also tried two different types of positive training examples. In the first set of experiments, we used only the domain homologs found by the *SAM-T98* method itself as a training set for each HMM. Thus, we simply used the *SAM-T98* models as they were given in the existing library of models. In the second experiment, we retrained each of these models using all of the data in (2) above. That is, using all of the SCOP sequences in the superfamily being modeled (but not in the family itself), and all of the domain homologs found by the given *SAM-T98* model and by the models built from other guide sequences from this superfamily (but not in the family itself). Thus in this

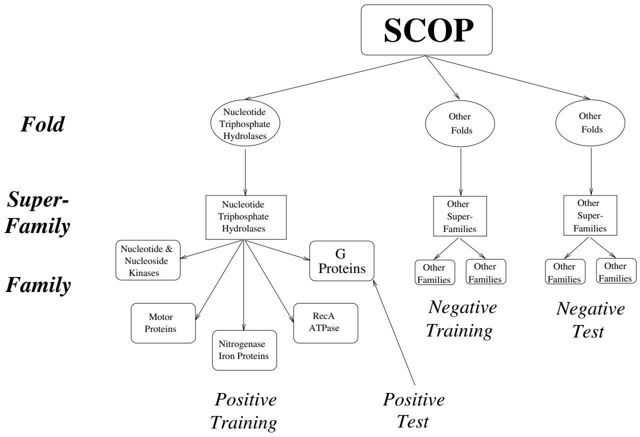

Figure 1: Separation of the SCOP PDB90 database into training and test sequences, shown for the *G proteins* test family.

latter case, each HMM was trained on same set of positive training examples used by the SVM-Fisher and BLAST:SCOP+SAMT98-homologs methods. Performance was somewhat better in the latter case at higher rates of false positives (RFP, see below), but was worse at lower RFP, making the method of less practical value. Therefore, we report the results of the first experiment here.

Results

Here we provide a comparison of the results of the best performing approaches for each of the methods. Since the numeric scores produced by each method are not directly comparable, we use the rate of false positives (*RFP*) achieved for each positive test sequence as metric for comparing methods (Park *et al.* 1998). The *RFP* for a positive test sequence is defined as the fraction of negative test sequences that score as good or better than the positive sequence.

G-proteins

Here, as an example, we look at the results for the *G proteins* family of the *nucleotide triphosphate hydrolases* SCOP superfamily.

The HMMs used in the recognition of members of the *G proteins* family were taken from two other families in the superfamily: *nucleotide and nucleoside kinases*, and *nitrogenase iron protein-like*. The positive training examples were the SCOP PDB90 sequences from the other families in the superfamily, along with the HMM domain homologs for the models for the guide sequences.

This experiment tested the ability of the methods

Sequence	BLAST	B-Hom	S-T98	SVM-F
5p21	0.043	0.010	0.001	0.000
1guaA	0.179	0.031	0.000	0.000
1etu	0.307	0.404	0.428	0.038
1hurA	0.378	0.007	0.007	0.000
1eft(3)	0.431	0.568	0.041	0.051
1dar(2)	0.565	0.391	0.289	0.019
1tadA(2)	0.797	0.330	0.004	0.000
1gia(2)	0.867	0.421	0.017	0.000

Table 1: Rate of false positives for *G proteins* family. BLAST = BLAST:SCOP-only, B-Hom = BLAST:SCOP+SAMT-98-homologs, S-T98 = SAMT-98, and SVM-F = SVM-Fisher method.

to distinguish the 8 PDB90 *G proteins* from 2439 sequences in other SCOP folds. The results are given in table 1. It is seen that the *SVM-Fisher* method scores 5 of the 8 *G proteins* better than all 2439 negative test sequences, and gets a lower rate of false positives than the other methods on the other 3 sequences, with the exception of 1eft-3.

We summarize the performance of the four methods in recognizing this family by looking at two overall figures of merit. The first is the maximum RFP for any sequence in the family. Under this measure of performance, we get 0.867 for BLAST:SCOP-only, 0.568 for BLAST:SCOP+SAMT98-homologs, 0.428 for *SAM-T98*, and 0.051 for SVM-Fisher for the G-proteins family.

Since the maximum RFP can be dominated by a few outliers, which for some reason may be particularly hard for a method to recognize, we also consider the median RFP for the sequences in the family. To calculate the median RFP, we require only that at least half of the sequences in the family be recognized, and calculate the maximum RFP for these sequences only. Of course, for each method, different sequences may be included in this "easiest half" of the family. Under this measure of performance, we get 0.378 for BLAST:SCOP-only, 0.330 for BLAST:SCOP+SAMT98-homologs, 0.007 for *SAM-T98*, and 0.0 for SVM-Fisher for the G-proteins family.

Overall results

In Table 2 we give the performance of all four methods on each of the 33 protein families we tested, as measured by the maximum and median RFP. We also computed these statistics for the first and third quartile, and the relative performance of the four methods was similar (data not shown).

A graphical comparison of the overall results for the

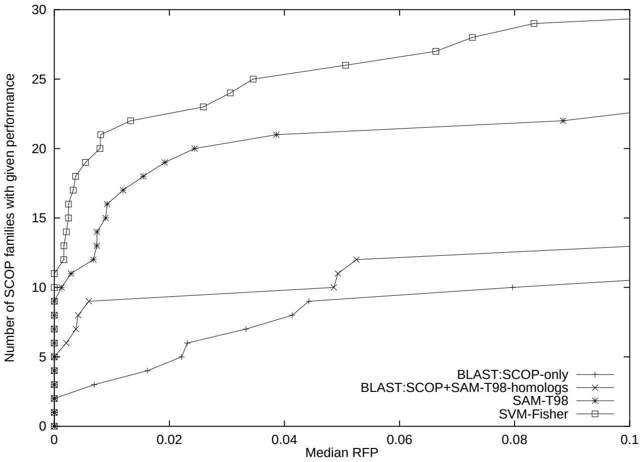

Figure 2: Here we compare the overall performance for the four methods on the 33 test families. For each family we computed the median RFP for that family, as shown in Table 2. Values for the median RFP are shown on the X-axis. On the Y-axis we plot the number of SCOP families, out of the 33 families that we tested, for which the given method achieves that median RFP performance or better.

33 test families focusing on low median RFP is given in figure 2.

Further experiments

We did further experiments to verify that the *SVM-Fisher* method was not relying too heavily on length and compositional bias in discriminating one protein domain family from another, as suggested by a referee of this paper. Such information would not be derivable from the test sequence in the case that the domain to be classified is contained in a larger test protein sequence. To simulate this situation, we appended randomly generated amino acids onto the ends of all the sequences in PDB90, creating a set of padded PDB90 sequences that all had length 1200 (the largest domain in PDB90 has length 905). The distribution of these random amino acids was determined from the overall amino acid frequencies in PDB90. The fraction of the padding that occured at the beginning of the sequence versus at the end of the sequence was determined uniformly at random as well.

We reran the experiments reported above with this padded PDB90 data set. In cases where homologs were used, these were randomly padded as well. Apart from a slight reduction in the amount of improvement over the other methods shown by *SVM-Fisher*, the results were on average qualitiatively similar to those obtained without padding. Details can be found in (Jaakkola,

#	Family	Maximum RFP				Median RFP			
		BLT	BLH	ST98	SVM	BLT	BLH	ST98	SVM
1	Phycocyanins	0.882	0.743	0.950	0.619	0.391	0.342	0.450	0.364
2	Long-chain cytokines	0.847	0.526	0.994	0.123	0.721	0.397	0.446	0.035
3	Short-chain cytokines	0.686	0.658	0.513	0.023	0.407	0.114	0.109	0.002
4	Interferons/interleukin-10	0.613	0.799	0.765	0.119	0.324	0.440	0.289	0.004
5	Parvalbumin	0.098	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6	Calmodulin-like	0.433	0.002	0.000	0.000	0.023	0.000	0.000	0.000
7	Immunoglobulin V dom	0.720	0.115	0.974	0.016	0.135	0.000	0.000	0.000
8	Immunoglobulin C1 dom	0.624	0.000	0.000	0.063	0.033	0.000	0.000	0.000
9	Immunoglobulin C2 dom	0.263	0.124	0.136	0.019	0.119	0.006	0.000	0.000
10	Immunoglobulin I dom	0.157	0.190	0.251	0.495	0.007	0.004	0.000	0.000
11	Immunoglobulin E dom	0.792	0.797	0.899	0.683	0.168	0.329	0.178	0.073
12	Plastocyanin/azurin-like	0.869	0.895	0.730	0.772	0.016	0.049	0.039	0.013
13	Multidomain cupredoxins	0.775	0.853	0.233	0.360	0.342	0.116	0.003	0.002
14	Plant virus proteins	0.975	0.940	0.782	0.410	0.641	0.391	0.088	0.133
15	Animal virus proteins	0.962	0.997	0.941	0.513	0.750	0.630	0.204	0.066
16	Legume lectins	0.551	0.895	0.643	0.552	0.278	0.298	0.278	0.083
17	Prokaryotic proteases	0.962	0.025	0.000	0.000	0.080	0.002	0.000	0.000
18	Eukaryotic proteases	0.846	0.000	0.000	0.000	0.000	0.000	0.000	0.000
19	Retroviral protease	0.500	0.195	0.183	0.187	0.238	0.108	0.012	0.003
20	Retinol binding	0.827	0.843	0.940	0.121	0.475	0.293	0.165	0.051
21	alpha-Amylases, N-term	0.935	0.953	0.737	0.037	0.630	0.851	0.007	0.000
22	beta-glycanases	0.974	0.939	0.370	0.079	0.517	0.338	0.009	0.008
23	type II chitinase	0.724	0.905	0.945	0.263	0.350	0.426	0.110	0.031
24	Alcohol/glucose dehydro	0.610	0.203	0.050	0.025	0.041	0.004	0.019	0.008
25	Rossmann-fold C-term	0.713	0.883	0.593	0.107	0.121	0.299	0.015	0.005
26	Glyceraldehyde-3-phosphate	0.791	0.537	0.062	0.004	0.315	0.102	0.009	0.002
27	Formate/glycerate	0.702	0.295	0.302	0.074	0.022	0.049	0.001	0.002
28	Lactate&malate dehydro	0.947	0.851	0.132	0.297	0.530	0.330	0.024	0.002
29	G proteins	0.867	0.568	0.428	0.051	0.378	0.330	0.007	0.000
30	Thioltransferase	0.205	0.072	0.986	0.029	0.000	0.000	0.000	0.000
31	Glutathione S-transfer	0.566	0.597	0.825	0.590	0.311	0.201	0.273	0.238
32	Fungal lipases	0.957	0.591	0.089	0.007	0.044	0.053	0.000	0.000
33	Transferrin	0.940	0.859	0.035	0.072	0.875	0.433	0.007	0.026

Table 2: Rate of false positives for all 33 families. BLT = BLAST:SCOP-only, BLH = BLAST:SCOP+SAMT-98-homologs, ST98 = SAMT-98, and SVM = SVM-Fisher method.

Diekhans, & Haussler 1998b). The datasets for all the experiments are available from our web site (Jaakkola, Diekhans, & Haussler 1998a).

Discussion

We have developed a new approach to the recognition of remote protein homologies that uses a discriminative method built on top of a generative model such as an HMM. Our experiments show that this method, which we call the SVM-Fisher method, significantly improves on previous methods for the classification of protein domains based on remote homologies.

All the methods considered in this paper combine multiple scores for each query sequence. The multiple scores arise either from several models that are available for a particular superfamily (HMM and SVM-Fisher) or because each known sequence can be scored against the query sequence with BLAST (Grundy 1998). The simple combination rules employed in this paper for each method are not necessarily optimal and further work needs to be done in this regard. It should be noted that while our methods for combining BLAST and HMM scores are essentially the same as those explored in (Grundy 1998), the relative performance of the simple generative HMM method versus the family pairwise search homology methods using BLAST is reversed in our experiments: here the HMM performs better. This is not surprising, since our tests consisted of finding very remote homologies for the most part, whereas the tests in (Grundy 1998) were for finding sequences that were mostly in the same family as the training sequences. Furthermore, the families in (Grundy 1998) were not defined by structure using SCOP, but rather by sequence similarity itself. There were also differences in the construction of the HMMs[1]. Our experimental results show, however, that it may be wise to build more powerful, discriminatively trained protein classification methods on top of HMM methods, rather than replace HMM methods with combinations of BLAST scores.

The discriminative SVM-Fisher method relies on the presence of multiple training examples from the superfamily of interest, and works best when these training sequences are not the same as those used to estimate the parameters of the underlying HMM. This presents us with an allocation problem, i.e., which sequences should be used for estimating the parameters of the HMM and which ones left for the discriminative method. This issue becomes especially important in cases where there are relatively few known sequences and homologs in the superfamily of interest. A possible solution to this problem and one that we have already successfully experimented with, concerns the use of generic protein models rather than those tuned to the particular family of interest. By generic models we mean HMMs constructed on the basis of statistical properties of short amino acid sequences that map on to structurally conserved regions in proteins (Bystroff & Baker 1997). Since the role of the HMM in our discriminative formalism is to provide features relevant for identifying structural relationships, the use of such generic models seems quite natural.

A generic model could also be trained as a single multi-way protein domain classifier, replacing the set of two-way classifiers we built for the experiments reported in this paper. It remains to be seen if this will be an effective way to construct a multi-way protein domain classifier, as compared to using some combination of the existing two-way classifiers. The lower rates of false positives achieved by our current two-way classifiers make us hopeful that an effective multi-way classifier can be built using some version of the SVM-Fisher method.

In the future, it will also be important to extend the method to identify multiple domains within large protein sequences. Since our experiments with artificially padded sequences were successful, we are confident that these methods can be adapted to the identification of multiple domains. However this work remains to be done. Finally, while this discriminative framework is specifically developed for identifying protein homologies, it naturally extends to other problems in biosequence analysis, such as the identification and classification of promoters, splice sites, and other features in genomic DNA.

[1] Our tests used the *SAM-T98* method for constructing HMMs, whereas the tests in (Grundy 1998) used an earlier version of the HMMER system (Eddy ; 1997), with the default parameters, which does not perform as well (Karchin & Hughey 1998); more recent and carefully tuned versions of HMMER would likely have performed better.

Acknowledgments

Tommi Jaakkola acknowledges support from a DOE/Sloan Foundation postdoctoral grant 97-2-6-CMB. David Haussler and Mark Diekhans acknowledge the support of DOE grant DE-FG03-95ER62112 and NSF grant IRI-9123692. We thank Richard Hughey and Kevin Karplus for suggestions and for making their software available.

References

Althschul, S., and Gish, W. 1996. Local alignment statistics. *Methods Enzymol.* 266:460–480.

Altschul, S.; Madden, T.; Schaffer, A.; Zhang, J.; Zhang, Z.; Miller, W.; and Lipman, D. 1997. Gapped

BLAST and PSI-BLAST: A new generation of protein database search programs. *NAR* 25:3899–3402.

Altshul, S. F.; Gish, W.; Miller, W.; W., M. E.; and J., L. D. 1990. Basic local alignment search tool. *JMB* 215:403–410.

Baldi, P.; Chauvin, Y.; Hunkapillar, T.; and McClure, M. 1994. Hidden Markov models of biological primary sequence information. *PNAS* 91:1059–1063.

Barrett, C.; Hughey, R.; and Karplus, K. 1997. Scoring hidden Markov models. *CABIOS* 13(2):191–199.

Brenner, S. E. 1996. *Molecular propinquity: evolutionary and structural relationships of proteins.* Ph.D. Dissertation, University of Cambridge, Cambridge, England.

Bystroff, C., and Baker, D. 1997. Blind predictions of local protein structure in casp2 targets using the i-sites library. *Proteins: Structure, Function and Genetics, Suppl.* 1:167–171.

Dubchak, I.; Muchnik, I.; Holbrook, S.; and Kim, S. 1995. Prediction of protein folding class using global description of amino acid sequence. *pnas* 92:8700–8704.

Dubchak, I.; Muchnik, I.; and Kim, S.-H. 1997. Protein folding class predictor for scop: Approach based on global descriptors. In *Proceedings, 5th International Conference on Intelligent Systems for Molecular Biology*, 104–108.

Durbin, R.; Eddy, S.; Krogh, A.; and Mitchison, G. 1998. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press.

Eddy, S. HMMER WWW site.
http://hmmer.wustl.edu/.

Eddy, S.; Mitchison, G.; and Durbin, R. 1995. Maximum discrimination hidden Markov models of sequence consensus. *J. Comput. Biol.* 2:9–23.

Eddy, S. 1995. Multiple alignment using hidden Markov models. In Rallings, C., et al., eds., *ISMB-95*, 114–120. Menlo Park, CA: AAAI/MIT Press.

Eddy, S. 1997. Hidden markov models and large-scale genome analysis. *Transactions of the American Crystallographic Association.* From a talk on HMMER and Pfam given at the 1997 ACA annual meeting in St. Louis.

Gish, W., and States, D. J. 1993. Identification of protein coding regions by database similarity search. *Nature Genetics* 3.

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

Grundy, W. N. 1998. Family-based homology detection via pairwise sequence comparison. In *Int. Conf. Computational Molecular Biology (RECOMB-98).* New York: ACM Press.

Henikoff, S., and Henikoff, J. G. 1994. Position-based sequence weights. *JMB* 243(4):574–578.

Hubbard, T.; Murzin, A.; Brenner, S.; and Chothia, C. 1997. SCOP: a structural classification of proteins database. *NAR* 25(1):236–9.

Hughey, R., and Krogh, A. 1995. SAM: Sequence alignment and modeling software system. Technical Report UCSC-CRL-95-7, University of California, Santa Cruz, Computer Engineering, UC Santa Cruz, CA 95064.

Hughey, R., and Krogh, A. 1996. Hidden Markov models for sequence analysis: Extension and analysis of the basic method. *CABIOS* 12(2):95–107. Information on obtaining SAM is available at http://www.cse.ucsc.edu/research/compbio/sam.html.

Jaakkola, T., and Haussler, D. 1998. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems 11.* San Mateo, CA: Morgan Kauffmann Publishers. to appear.

Jaakkola, T.; Diekhans, M.; and Haussler, D. 1998a. Data set for *a discriminative framework for detecting remote protein homologies.*
http://www.cse.ucsc.edu/research/compbio/discriminative/fisher-scop-data.tar.gz.

Jaakkola, T.; Diekhans, M.; and Haussler, D. 1998b. A discriminative framework for detecting remote protein homologies. Unpublished, available from http://www.cse.ucsc.edu/research/compbio/research.html.

Karchin, R., and Hughey, R. 1998. Weighting hidden Markov models for maximum discrimination. Bioinformatics to appear.

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

Karplus, K.; Kimmen Sjölander; Barrett, C.; Cline, M.; Haussler, D.; Hughey, R.; Holm, L.; and Sander, C. 1997. Predicting protein structure using hidden Markov models. *Proteins: Structure, Function, and Genetics* Suppl. 1:134–139.

Krogh, A.; Brown, M.; Mian, I. S.; Sjölander, K.; and Haussler, D. 1994. Hidden Markov models in computational biology: Applications to protein modeling. *JMB* 235:1501–1531.

Linial, M.; Linial, N.; Tishby, N.; and Yona, G. 1997. Global self-organization of all known protein sequences reveals inherent biological signatures. *jmb* 268(2):539–556.

Mamitsuka, H. 1996. A learning method of hidden markov models for sequence discrimination. *Journal of Computational Biology* 3(3):361–373.

Neuwald, A.; Liu, J.; Lipman, D.; and Lawrence, C. E. 1997. Extracting protein alignment models from the sequence database. *NAR* 25:1665–1677.

Park, J.; Karplus, K.; Barrett, C.; Hughey, R.; Haussler, D.; Hubbard, T.; and Chothia, C. 1998. Sequence comparisons using multiple sequences detect twice as many remote homologues as pairwise methods. *JMB* 284(4):1201–1210. Paper available at `http://www.mrc-lmb.cam.ac.uk/genomes/jong/assess_paper/assess_paperNov.html`.

Pearson, W., and Lipman, D. 1988. Improved tools for biological sequence comparison. *PNAS* 85:2444–2448.

Rabiner, L. R., and Juang, B. H. 1986. An introduction to hidden Markov models. *IEEE ASSP Magazine* 3(1):4–16.

Taylor, W. 1986. Identification of protein sequence homology by consensus template alignment. *JMB* 188:233–258.

WU-BLAST WWW archives. `http://blast.wustl.edu/`.