Protein Homology Detection with Biologically Inspired Features and Interpretable Statistical Models

Pai-Hsi Huang* and Vladimir Pavlovic

Department of Computer Science,
Rutgers University, Piscataway, NJ 08854-8019, U.S.A.
E-mail: paihuang@cs.rutgers.edu E-mail: vladimir@cs.rutgers.edu
*Corresponding author

Abstract: Computational classification of proteins using methods such as string kernels and Fisher-SVM has demonstrated great success. However, the resulting models do not offer an immediate interpretation of the underlying biological mechanisms. In particular, some recent studies have postulated the existence of a small subset of positions and residues in protein sequences may be sufficient to discriminate among different protein classes. In this work, we propose a *hybrid* setting for the classification task. A generative model is trained as a feature extractor, followed by a *sparse* classifier in the extracted feature space to determine the membership of the sequence, while discovering features relevant for classification. The set of *sparse* biologically motivated features and the discriminative method offer the desired biological interpretability. We apply the proposed method to a widely used dataset and show that the performance of our models is comparable to that of the state-of-the-art methods. The resulting models use *fewer than* 10% of the original features. At the same time, the sets of critical features discovered by the model appear to be consistent with confirmed biological findings.

Keywords: sequence classification, homology detection, discriminative learning, biologically motivated features, feature selection.

Reference to this paper should be made as follows: Pai-Hsi Huang and Vladimir Pavlovic (xxxx) 'Protein Homology Detection with Biologically Inspired Features and Interpretable Statistical Models', *Int. J. of Data Mining and Bioinformatics*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Vladimir Pavlovic is an Assistant Professor in the Computer Science Department at Rutgers University. He received the PhD in electrical engineering from the University of Illinois in Urbana-Champaign in 1999. From 1999 until 2001 he was a member of research staff at the Cambridge Research Laboratory, Cambridge, MA. Before joining Rutgers in 2002, he held a research professor position in the Bioinformatics Program at Boston University. Vladimir's research interests include time-series modeling, statistical computer vision and bioinformatics. His research group is based at the Sequence Analysis and Modeling (SEQAM) Lab, http://seqam.rutgers.edu.
Pai-Hsi Huang is a PhD candidate in the Computer Science Department at Rutgers University. He received his masters degree in Statistics in 2005 and in Computer Science in 2004 from Rutgers University. He also

serves as the president of an academic club, Rutgers Graduate Bioinformatics Association. Pai-Hsi's research interest include statistical analysis and modeling, sequence analysis and bioinformatics. Pai-Hsi is currently a member of the SEQAM lab, under the supervision of Professor Pavlovic.

1 Introduction

Protein homology detection is a fundamental problem in computational biology. With the advance of large-scale sequencing techniques, it becomes evident that experimentally determining the function of an unknown protein sequence is an expensive and tedious task. Currently, there are more than 54 million DNA sequences in GenBank (Benson et al. (2005)), and approximately 208,000 annotated and 2.6 million unannotated sequences in UNIPROT (Bairoch et al. (2005)) . The rapid growth of sequence databases makes development of computational aids for functional annotation a critical and timely task.

Early approaches to computationally-aided homology detection, such as BLAST (Altschul et al. (1990)) and FASTA (Pearson and Lipman (1988)), rely on aligning the query sequence to a database of known sequences (pairwise alignment). However, the weakness of the pairwise approach is its lack use of data: alignment is performed on the query sequence to each of the sequences in the database *one at a time*. Later methods, such as profiles (Gribskov et al. (1987)) and profile hidden Markov models (profile HMM) (Eddy (1998)) collect aggregate statistics from a group of sequences known to belong to the same family. Upon query time, an unknown sequence is aligned to all models to see if there is a significant *hit*. Profile HMMs have demonstrated great success in protein homology detection. The linear structure of a profile HMM offers great interpretability to the underlying process that generates the sequences: the *match* states represent positions in the superfamily that are *conserved* throughout the evolutionary process. However, as *generative* models, profile HMMs are estimated from sequences known to belong to the same superfamily and do not attempt to capture the differences between members and non-members. Also, it has been shown that profile HMMs are unable to detect members with low sequence identity.

To tackle these deficiencies, Jaakkola et al. (1999) proposed *SVM-Fisher*. The idea is to combine a generative model (profile HMM) with a discriminative model (support vector machines, SVM) and perform homology detection in two stages. In the first stage, the generative model, *trained using positive sequences only*, extracts features from all sequences (*positive and negative*). In the second stage, with the fixed-length features, the discriminative model constructs the decision boundary between the two classes.

The class of *string kernels*, on the other hand, bypasses the first stage and directly model the decision boundary using SVMs. The *spectrum kernel* (Leslie et al. (2002a)), the *mismatch kernel* (Leslie et al. (2002b)) and the *profile kernel* (Kuang et al. (2004)) define different notions of *neighborhood* for a subsequence of size $k \geq 1$ and determine the similarity between the two sequences as a function of the size of the intersection of their neighborhood.

Previous studies showed that both approaches, the *SVM-Fisher* approach, and the class of *string kernels*, are more effective than the generative models[a]. Despite their great success, these two approaches are not readily interpretable or, when an interpretation of the models is available, it may not be biologically intuitive. For instance, the model should be able to explain how sequences in the same superfamily evolve over time. Are there certain positions that are *critical* to a superfamily? If so, what kind of physical/chemical properties should such positions possess? Although profile HMMs attempt to offer such explanations but as generative models they lack the discriminative interpretability.

The central idea of our work is to develop an interpretable method for protein homology detection. Our approach is motivated by the results presented in Kister et al. (2002); Reva et al. (2002); Kister et al. (2001) that postulate the existence of a small subset of positions and residues in protein sequences may be sufficient to discriminate among different protein classes. We aim to recover these *critical positions* and the type of residues that must occur at these positions using a new set of features embedded in a class of discriminative models. The combination of the features and the classifier may offer a *simple and intuitive* interpretation to the underlying biological mechanism that generates the biosequences.

2 Related works

Denote X as a protein sequence. Jaakkola et al. (2000, 1999) proposed to use the gradient of the log-likelihood of the sequence, X, with respect to the model parameters as features:

$$\begin{aligned} f_{\tilde{x},\tilde{s}} &= \frac{\partial}{\partial \theta_{\tilde{x},\tilde{s}}} log P(X|\Theta) \\ &= \frac{\xi(\tilde{x},\tilde{s})}{\theta_{\tilde{x}|\tilde{s}}} - \xi(\tilde{s}), \end{aligned} \tag{1}$$

where $\tilde{\mathrm{x}} \in \Sigma$, the alphabet set, $\tilde{\mathrm{s}} \in S$, the emitting states in the model, Θ represents the set of parameters of the model, $\theta_{\tilde{x},\tilde{s}}$ represents the emission probability of symbol $\tilde{\mathrm{x}}$ at state $\tilde{\mathrm{s}}$, and $\xi(\tilde{x},\tilde{s})$ as well as $\xi(\tilde{s})$ are the sufficient statistics, obtained using the forward-backward algorithm in Rabiner (1990):

$$\begin{aligned} \xi(\tilde{x},\tilde{s}) &= \sum_{t=1}^{T_X} P(S_t = \tilde{s}, X_t = \tilde{x}|X,\Theta) \\ &= \sum_{t=1}^{T_X} P(S_t = \tilde{s}|X,\Theta) I(X_t = \tilde{x}), \end{aligned} \tag{2}$$

where T_X is the length of X, S_t is the state that is traversed at time t, X_t is the t^{th} symbol of X, $1 \leq t \leq T_X$, and $I(.)$ denotes the indicator function. The extracted fixed-length features are referred to as the *Fisher scores* and used to build the SVM for superfamily classification. Feature dimensionality can be further reduced via

[a]Such results are demonstrated in Jaakkola et al. (1999, 2000); Leslie et al. (2002b,a); Kuang et al. (2004); Interested readers may want to refer to the cited papers for more details.

the *9-component Dirichlet mixture prior*, proposed by Sjolander et al. (1996). The SVM-Fisher approach has received some criticism because an inference procedure of quadratic complexity is required for each sequence[b]. Although the criticism does address a valid concern for a general HMM, in the case of a profile HMM, such issue does not exist: the linear structure enables one to make inference in linear time.

The methods based on *string kernels*, on the other hand, bypass the need of a generative model as a feature extractor. Given a sequence, X, the *spectrum-k* kernel (Leslie et al. (2002a)) first *implicitly* maps it to a d-dimensional vector, where $d = |\Sigma|^k$:

$$\Phi_k(X) = \sum_{\alpha} (I(\alpha = \gamma))_{\gamma \in \Sigma^k}. \quad (3)$$

Next, the similarity between X and Y is then defined as:

$$K(X, Y) = \Phi_k(X)^T \Phi_k(Y), \quad (4)$$

where in Eq. (3) α denotes all *k-mers* in X and γ denotes a member in the set of all *k-mers* induced by Σ, the alphabet set. The $mismatch(k, m)$ kernel (Leslie et al. (2002b)) relaxes exact string matching by allowing up to m mismatches between α and γ. In such setting, each element in the kernel matrix takes $O(k^{m+1}|\Sigma|^m(T_X + T_Y))$ time to compute.

3 Proposed features and methods

Our computational approach to remote homology detection involves two steps: feature extraction with dimensionality reduction followed by joint classification and feature selection in the constructed feature space. A crucial aspect of this approach lies in the ability to impose the sparsity constraint, which leads to significant reduction in the number of utilized features as well as the interpretability of the final model. We show the proposed *hybrid* procedure in Fig. 1.

3.1 Feature extraction and dimensionality reduction

We use the sufficient statistics of the sequences with respect to the profile HMM as features. This choice of features may allow immediate biological interpretation of the constructed model. In particular, we use the sufficient statistics that are associated with the the symbols of the *match* states. We focus only on the match states because the structure of a profile HMM indicates that these states represent the positions that are conserved throughout evolution. These features can be obtained using Eq. (2) with

$$P(S_t = \tilde{s}|X, \Theta) = \frac{\alpha_{\tilde{s}}(t)\beta_{\tilde{s}}(t)}{P(X|\Theta)} \quad (5)$$

[b]For a general HMM exhibiting no special structure, the complexity of the inference procedure takes $O(n^2T)$ time, where n is the number of states in the HMM and T is the length of the sequence (Rabiner (1990)).

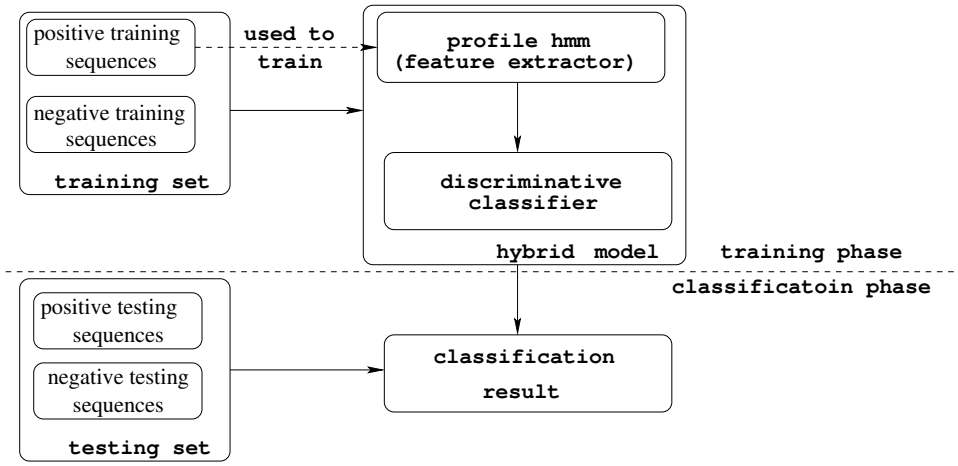


Figure 1 A schematic depiction of our hybrid model.

where $\alpha_{\tilde{s}}(t)$ and $\beta_{\tilde{s}}(t)$ are the forward and backward probabilities defined in Rabiner (1990). In this setting, each example is represented by a vector of length $d = m|\Sigma|$, where m is the number of match states in the profile HMM and $|\Sigma| = 20$. To reduce dimensionality, we partition all 20 amino acids into the following four groups, according to their chemical and physical properties:

- Group 1 – Non polar, hydrophobic: {F, M, W, I, V, L, A, P}.
- Group 2 – Negatively charged, polar, hydrophilic: {D, E}.
- Group 3 – No charge, polar, hydrophilic: {C, N, Q, T, Y, S, G}.
- Group 4 – Positively charged, polar, hydrophilic: {H, K, R}.

As a result, we represent each example by the following:

$$f_{g,\tilde{s}} = \sum_{\tilde{x}\in\Sigma} \xi(\tilde{x}, \tilde{s}) I(\tilde{x} \in \text{Group } g), \tag{6}$$

where $g \in \{1, 2, 3, 4\}$ represents each group of amino acid. The partition reduces the dimensionality, d, from $20m$ to $4m$, compared to $9m$ in Jaakkola et al. (2000, 1999). Our experiments in Sec. 4 confirm the effectiveness of this representation [c].

3.2 Classification and Feature Selection via Logistic Regression

Let f_i be the features extracted from the i^{th} sequence, X_i, and $c_i \in \{0, 1\}$ be the response variable, where $c_i = 1$ denotes membership of the superfamily. The logistic regression model defines the probability of sequence X_i belonging to the superfamily of interest, $\pi_i = P(c(X_i) = 1)$, as:

$$\pi_i = \phi(\beta^T f_i) \quad = \quad \frac{exp(\beta^T f_i)}{1 + exp(\beta^T f_i)} \tag{7}$$

[c] We have also performed dimensionality reduction using the 9-component Dirichlet priors and do not notice any significant difference in performance.

where β is the parameter of the model and $\phi(.)$ is the *cumulative distribution function* (CDF) of a logistic distribution. To estimate the parameters of the logistic model one sets $\hat{\beta}$, the estimate, to β^*, where β^* is the parameter vector that maximizes the following objective function, which is also the joint likelihood function of the observed data:

$$J(\beta) = \prod_{i=1}^{n} \pi_i^{c_i}(1-\pi_i)^{(1-c_i)}, \tag{8}$$

where π_i is the probability of sequence X_i having class c_i as its label, and π_i is a function of β. There are existing algorithms for estimating β, such as *Iteratively Reweighted Least Squares* algorithm. Like SVM, the logistic model is also a *discriminative* classifier.

3.3 Interpretation of the logistic model with the proposed features

Use of the logistic model provides a simple and intuitive description of data. If the assumption, $p(c = 1|\mathbf{f}, \beta) = \phi(\mathbf{f}^T\beta)$, holds, then the contribution of each predictor variable, $f^{(j)}$, $1 \leq j \leq d$, is reflected in the corresponding model parameter, β_j. A coefficient with a large absolute value implies that the corresponding position has a strong preference for a type of amino acids: the position prefers a specific group of amino acids to be present when the coefficient is large and positive and prefers a specific group of amino acids to be absent when the coefficient is large and negative.

Moreover, β also offers a probabilistic interpretation. Define the *odds* of an event with probability p of occurring as $\frac{p}{1-p}$; given the estimated parameter $\hat{\beta}$, and a feature vector f_i representing sequence X_i in the *feature space*, the estimated *odds* of sequence X_i belonging in the superfamily is:

$$odds(X_i \in \ supFam) = \frac{\hat{\pi_i}}{1-\hat{\pi_i}} = exp(\hat{\beta}^T f_i). \tag{9}$$

Define a new sequence $X_{i'}$ such that $f_{i'}^{(k)} = f_i^{(k)}$, $\forall 1 \leq k \leq d$, except $f_{i'}^{(j)} = f_i^{(j)} + 1$, for one specific j, $1 \leq j \leq d$, meaning we increase $f_i^{(j)}$, the j^{th} covariate of example i, by one unit. In this case, the estimated odds of the new sequence $X_{i'}$ is:

$$\begin{aligned} odds(X_{i'} \in \ supFam) &= exp(\hat{\beta}^T f_i + \hat{\beta}_j) \\ &= \frac{\hat{\pi_i}}{1-\hat{\pi_i}} exp(\hat{\beta}_j). \end{aligned} \tag{10}$$

Equation (10) indicates that the odds are multiplied by $exp(\hat{\beta}_j)$ when we increase the j^{th} covariate of example i by one unit. For example, suppose at position $\tilde{s}$, the corresponding parameter $\hat{\beta}^{\tilde{s},\tilde{x}}$ for symbol $\tilde{x}$ is $0.1615 = log(1.175)$. Then the odds of a sequence, X, being in the superfamily increases by 17.5 percent if in X, the symbol $\tilde{x}$ aligns to the model at position $\tilde{s}$.

One may argue that the preference for presence or absence of a specific group of amino acids at a position in a group of sequences is already reflected in the profile HMM and using a logistic model to recover the desired information is redundant.

However, this need not be the case: one position in a specific superfamily may prefer a certain group of amino acids which is also preferred by another group of sequences. In this case the corresponding coefficient in the logistic model we proposed will be *insignificant*, close to 0. A coefficient corresponding to a certain type of amino acids at one position will be significant if, for example, it has been observed that the group of amino acids are present in the family of interest (the positive examples) and are absent in all the other families (the negative examples).

3.4 Use of Sparsity-enforcing Regularizers

It is well known in the statistical learning community that when the provided positive examples and negative examples are indeed *separable*, then the objective function in Eq. (8) is *unbounded* and there exist infinitely many solutions. As a result, some type of *regularization* on β is preferred. Performing regularization on β can be interpreted as placing a *prior distribution* on β under the *Bayesian learning paradigm*.

Our belief that the model may be *sparse* leads us to set the prior distribution $\beta \sim N(0, A)$, where A is some covariance matrix. In our study, we set A to be some *diagonal* matrix and the induced objective function becomes the *posterior distribution* of β:

$$J(\beta)_{Gaussian} \propto e^{-\frac{1}{2}\|\beta\|_2} \cdot \prod_{i=1}^{n} \pi_i^{c_i}(1-\pi_i)^{(1-c_i)}, \tag{11}$$

where $\|\cdot\|_k$ denotes the $l-k$ norm of a vector. Such an assignment states that all β_is are *mutually independent*. The independence assumption is clearly violated, since the features that we use are sufficient statistics. However, it is impractical to assume a general covariance structure for β, as one will need to either specify or estimate the $\binom{d}{2}$ parameters in advance. On the other hand, Gaussian priors often do not set the coefficients corresponding to the irrelevant features to 0, because the shape of the distribution is too mild around the origin. Therefore, we also use priors that *promote* and *enforce* sparsity such as the Laplace priors. In such setting, we assume that $\beta_i \sim N(0, \tau_i)$, for $1 \leq i \leq d$. Furthermore, we place a *hyper prior*, γ, on every τ_i:

$$p(\tau_i|\gamma) = \frac{\gamma}{2} e^{-\frac{\gamma\tau_i}{2}}. \tag{12}$$

Integrating out every τ_i, we have

$$p(\beta_i|\gamma) = \frac{\sqrt{\gamma}}{2} e^{-\sqrt{\gamma}|\beta_i|}. \tag{13}$$

And the induced objective function, still a *posterior distribution* of β becomes:

$$J(\beta)_{Laplacian} \propto e^{-\|\beta\|_1} \cdot \prod_{i=1}^{n} \pi_i^{c_i}(1-\pi_i)^{(1-c_i)}. \tag{14}$$

The hierarchical model shows that each β_i now, follows a Laplace distribution. The Laplacian priors produce *sparser* models than Gaussian priors. We plot the density functions of a standard Gaussian (solid line) and a standard Laplacian (broken line) distributions in Fig. 2.

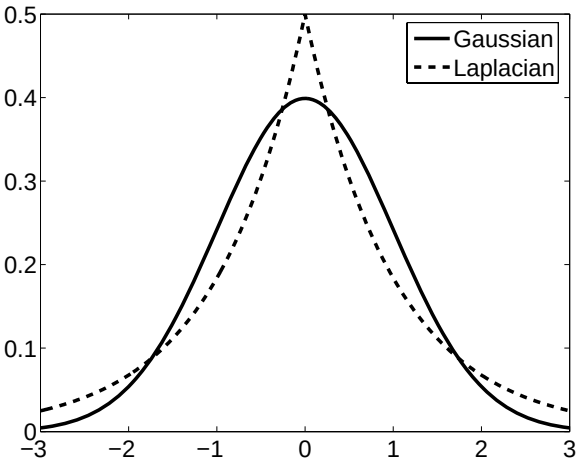


Figure 2 The density functions of a standard Gaussian (solid line) and a standard Laplacian (broken line) distributions.

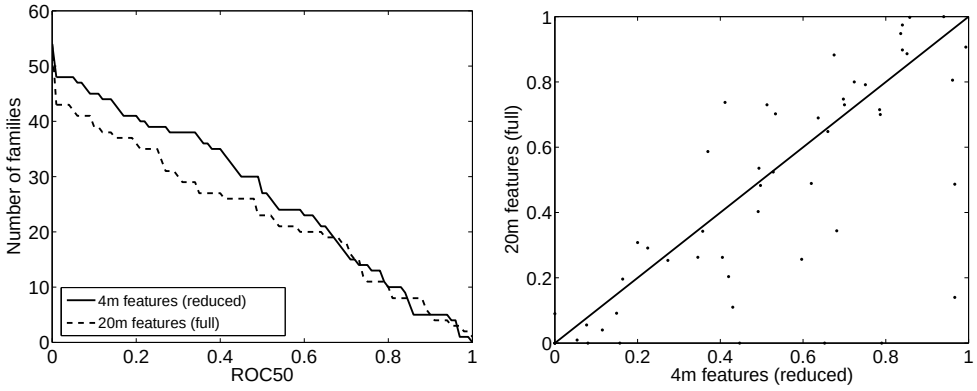
3.5 A Similar Setting with SVM

Given the feature vectors, one may also build the decision boundary using an SVM. In the case of a linear kernel, like the logistic model, the SVM also builds a linear decision boundary to discriminate between the two classes. However, the results produced by an SVM is interpretable *only* when a *linear* (or possibly *polynomial*) kernel is employed. While the objective functions in an SVM setting and a logistic regression settings are different, the results are often similar.

4 Experiments and Results

We use the dataset published in Kuang et al. (2004) to perform our experiments. The dataset contains 54 target families from SCOP 1.59 (Lo Conte et al. (2000)) with 7329 SCOP domains. No sequence shares more than 95% identity with any other sequence in this dataset, as indicated in Kuang et al. (2004). This dataset has a history and its variants have been used as a gold standard for protein remote homology detection in various studies. Sequences in the SCOP database are domains extracted from proteins in the Protein Data Bank (Berman et al. (2000)), which is a centralized repository for proteins with known three dimensional structure. Sequences in SCOP are placed in a tree-like hierarchy. Proteins in the same family clearly share a common evolutionary origin; proteins in the same superfamily have low sequence similarity but it is very likely that they share a common evolutionary origin. Proteins in the same fold share similar secondary structure in the same arrangement and with the same topological connections, but need not share a common evolutionary origin. Remote homology detection means classification on the superfamily level.

Jaakkola et al. (2000, 1999) proposed the following setup for the experiments. Suppose a superfamily S^i is under fold F^j and suppose S^i has k families, pick the sequences in $k-1$ families as the *positive training* sequences and the sequences in



(a) (b)

Figure 3 Comparison of performance of the full and reduced feature sets. The classifier used here is the logistic classifier with Normal prior. Panel (a) shows the number of families whose ROC-50 scores are better than a given threshold for the sets of full and reduced features. Panel (b) depicts the pairwise scatter-plot of ROC-50 scores for the two classifiers utilizing these two sets of features.

the left-out family will be used as the *positive testing* sequences. Negative training and testing sequences come from two different folds F^k and F^l, $k \neq l$, $k \neq j$, $l \neq j$, to avoid giving the classifier unnecessary advantage. All sequences in fold F^j but not in superfamily S^i are not used because their relationship with the *target* superfamily, S^i, is uncertain, as suggested in Jaakkola et al. (2000, 1999). In subsequent studies, such as Liao and Noble (2002); Leslie et al. (2002a,b); Kuang et al. (2004), different versions of the database are used as more sequences are deposited into the database.

We evaluate all methods using the *Receiver Operating Characteristic* (ROC) and ROC-50 (Gribskov and Robinson (1996)) scores. The ROC-50 score is the (normalized) area under the ROC curve computed up to 50 false positives. With small number of positive testing sequences and large number of negative testing sequences, the ROC-50 score is more indicative of the prediction accuracy of a homology detection method.

All profile HMMs for our *hybrid* procedure are obtained in the following way: first, we locate the profile most suitable for the experiment and download the multiple alignment from PFam (Bateman et al. (2004)); next, we estimate an initial profile HMM from the multiple alignment; finally, we refine the profile HMM using the labeled positive training sequences in the dataset. We use an algorithm similar to the Expectation-Maximization (EM) algorithm (Dempster et al. (1977)) to refine the profile HMM with the 9-component mixture of Dirichlet priors (Sjolander et al. (1996))[d]. To avoid over-representation of sequences, we also incorporate *position-based* sequence weighing scheme (Henikoff and Henikoff (1994)). Once a profile HMM for the superfamily of interest is estimated, we use it to extract *fixed-length*

[d]With mixture of Dirichlet priors, the Maximization step can no longer be performed using closed-form solutions. As a result, to speed up estimation, instead of obtaining the *posterior mode*, we obtain the *posterior mean*. Typically, the likelihood of the observed data increases up three digits of precision after the decimal point. Then the algorithm starts bumping around some mode.

features, the sufficient statistics with respect to the emission probabilities of the *match* states, and we use the extract features to train the *discriminative* classifier, in our case, the logistic regression model.

For logistic models, we perform our experiments on Normal and Laplace priors using *Bayesian Binary Regression Software* (BBR) (Genkin et al. (pear)). Precision γ in the Laplace models are set to the value suggested by Genkin et al. (pear). Experiments using linear kernel SVM make use of an existing machine-learning package called *Spider* (available at *http://www.kyb.tuebingen.mpg.de/bs/people/spider*).

In Fig. 3, we compare the performance of the full and reduced feature sets. The classifier used is the logistic classifier with Normal prior. The two sets of features perform similarly with SVM (linear kernel) and therefore are not reported. The dimensionality of the full feature set is $|\Sigma|m = 20m$, where $|.|$ denotes the cardinality and m denotes the number of *match* states; the dimensionality of the reduced features is $4m$. Fig. 3(a) shows the number of families (vertical axis) achieving a corresponding ROC-50 score (horizontal axis) for the two sets of features. It appears that the performance of the two sets of features are comparable, although in the area of low ROC-50 score, the set of reduced features seems to perform better, implying higher prediction accuracy. Fig. 3(b) shows the pairwise scatterplot of the ROC-50 scores for these two sets of features. A point falling under the diagonal line in the figure represents a case in which the reduced feature set achieves better performance. Out of 54 experiments, 28 and 21 of them fall under and above the diagonal, respectively. The p-value of the sign test is 0.39, indicating no strong evidence to support the claim that dimensionality reduction degrades the performance. In all subsequent reports, all logistic models use the reduced feature set.

In Fig. 4, we compare the performance of different methods. Fig. 4(a) and Fig. 4(b) indicate that, with ROC-50 score greater than 0.4, both logistic models (Normal and Laplacian priors) dominate the mismatch kernel. Furthermore, the performance of both logistic models appears to be comparable in the area of high ROC-50 score (> 0.8); but in the area of low ROC-50 score, the logistic model with Laplacian prior shows slightly higher prediction accuracy. Finally, *SVM-Fisher* performs well in the area of high ROC-50 score; however, the performance starts to degrade when ROC-50 score falls under 0.8. In Fig. 4(c), points falling above the diagonal corresponds to an experiment in which the logistic model with Normal prior performs better than the mismatch(5,1) kernel. Out of 54 experiments, 30 and 22 points fall above and under the diagonal, respectively, resulting in a p-value of 0.33 indicating no strong evidence to conclude which one of the two methods performs better. Likewise, in Fig. 4(d), 25 and 26 points fall above and under the diagonal line, suggesting that the performance of the logistic model with a Laplacian prior is comparable to that of a Normal prior.

We summarize the mean ROC and ROC-50 score of different methods for a quick reference and comparison in Tab. 1.

4.1 The Sparse Model

Enforcing sparsity in the number of parameters can be viewed as a *feature selection* process. The logistic model with Laplacian prior discards the irrelevant features by setting the corresponding parameters to 0. Among 54 families, there

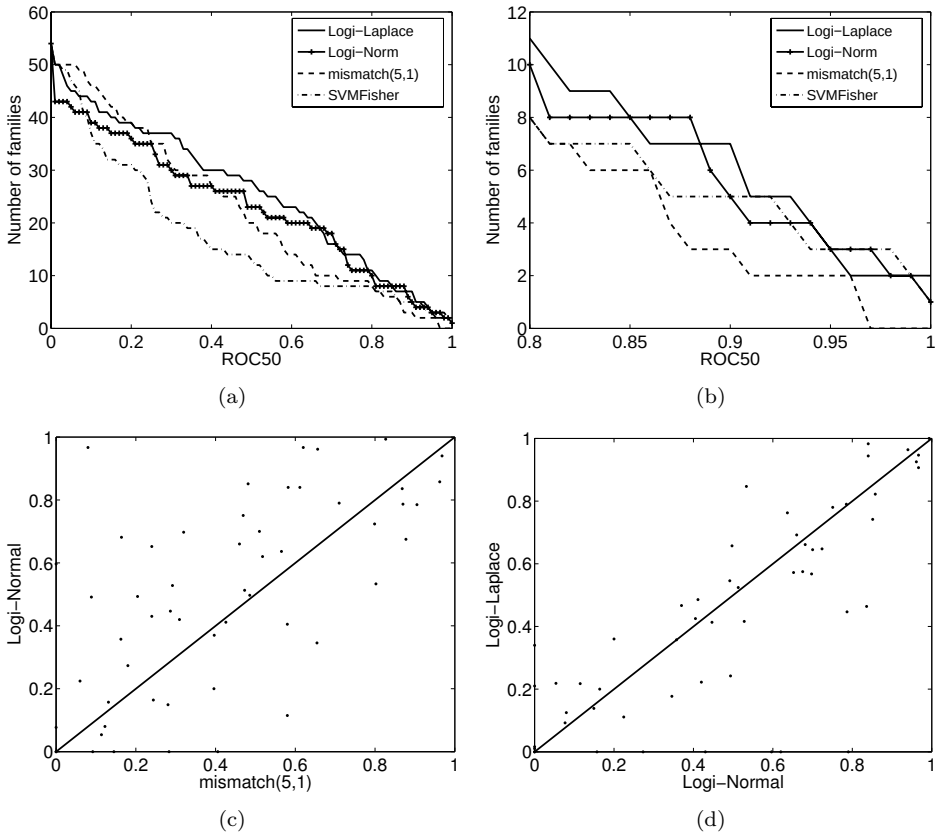


Figure 4 Comparison of performance of mismatch(5,1) kernel, SVM-Fisher, and logistic model with Normal and Laplacian priors. Panel (a) shows the number of families whose ROC-50 scores are better than a given threshold. Panel (b) shows the detail plot of the high ROC-50 score region of (a). Panel (c) shows the pairwise scatter-plot of ROC-50 scores for the logistic model with Normal prior and the mismatch(5,1) kernel. Panel (d) shows the pairwise scatter-plot of ROC-50 scores for the logistic models with Normal and Laplace priors.

are, on average, 480 features to select from. The Laplacian prior selects only about 43 features per family, resulting in more than 90% reduction in the final number of selected features. At the same time, the performance of the model with the reduced feature set remains indistinguishable from that of the model with a full feature set.

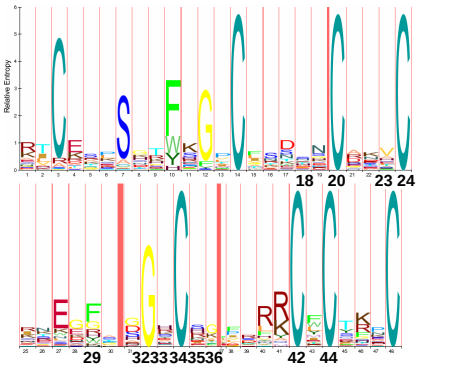
The set of features selected by the sparse model can offer interesting insights into the biological significance of the discovered "critical positions". For example, our experimental results indicate that the performance of this class of classifiers is good and consistent on the *Scorpion toxin-like* superfamily. In one particular family, *Plant defensins*, out of 188 features[e], the logistic model with Laplacian prior selects 19 features, scattered on approximately 12 positions. The ROC-50 score of the classifier on this family is 1. Upon further investigation, we extract these *critical positions* along with their preferred symbols: {(18[18], **E**), (20[20], **C**), (23[23], **H**), (24[24], **C**), (29[29], **G**), (34[32], **G**), (35[33], **K/Y**), (36[34], **C**), (37[35], **D/Y**),

[e]This corresponds to 47 positions, since each position is represented by 4 features.

Table 1 Mean ROC and ROC-50 scores for different homology detection methods

	mean ROC score	mean ROC-50 score
Logistic-Normal(4m)	.883256	.491564
Logistic-Laplace(4m)	.847313	.438070
Logistic-Normal(20m)	.813895	.426925
Logistic-Laplace(20m)	.864500	.474170
mismatch(5,1)	.874890	.416650
SVM-Fisher	.756618	.319048

(38[36], **G/N**), (41[42], **C**), (43[44], **C**)}, where in each pair, the leading number corresponds to the position in our profile HMM, the number in the bracket corresponds to the position in the HMM-logo in Fig. 5(a), and the letter the preferred symbol at that position. The positions slightly disagree because we use a different heuristic to determine whether a column in a multiple alignment corresponds to a *match* state or an *insertion* state. We also show the schematic representation of this family suggested by the *PROSITE database* Hulo et al. (2006) in Fig. 5(b); in this figure, each symbol 'C' represents a conserved cysteine involved in a disulphide bond. Hence, our classifier captures some of the conserved cysteine residues: (20,



(a) (b)

Figure 5 Panel (a): The HMM-logo of the *plant defensins* family. We obtain this logo from PFam. Panel (b): The schematic representation of this family suggested by PFam and PROSITE.

24, 34, 41 and 43), with the preferred symbols, that are *critical for discerning* plant defensins from other similar proteins. Other conserved cysteins are not selected as critical features. Upon detailed examination, it became clear that while conserved in Plant defensines, these positions also appeared to be conserved in other similar families with the same symbol and were, therefore, not deemed discriminative. A set of 9 different residues seemed to play a more critical classification role.

Next, we extract the *positive* coefficients from the estimated β for the logistic model and plot the weights in Fig. 6(a). It appears that there are 3 *critical regions* for this family: positions $18-29$, $33-39$ and $40-44$. We obtain the primary and secondary structure, in schematic diagrams, of 8 sequences belonging to this family from the *PDBsum database* Laskowski et al. (2005). It appears that there are some common secondary structure among these 8 sequences: a $\beta-$*strand* occurs in the neighborhood of positions $3-7$, a *helix* occurs in the neighborhood of positions $18-28$ and two $\beta-$*strands* occur in the neighborhood of positions $32-38$ and $41-48$, respectively. There seems to be some correlation between these *conserved* secondary structures and the *critical regions* indicated by our logistic model with Laplacian prior. We are currently performing further analyses to understand the connection between them.

We obtain similar results for another family: *Short-chain scorpion*. The ROC-50 score of the classifier for this family is 0.91. We show the HMM-logo for this family in Fig. 7(a) and the schematic representation of this family in Fig. 7(b). Out of 116 features in this family, the sparse classifier selects 14 of of them: (1, **C**), (4, **N**), (7, **C**), (11,**C**), (15, **G**), (17, **A**), (18, **S**), (19, **G/S**), (20, **G**), (21, **Y**), (22, **C**), (24, **G**), (27, **C**), (29, **C**). It appears that our sparse model captures all conserved cysteine residues in this family, indicating that, unlike in the family *Plant defensins*, these conserved cysteine residues are *unique* to this family. On the other hand, the conserved lysine (K) residue at positions 12, 21, and 26 in Fig. 7(a) are deemed *insignificant* by our sparse model because sequences not belonging to this family also appear to have such residues aligned to these positions.

Finally, it is worth noting that the dense model with Normal priors also achieved classification performance similar to the sparse model. However, the weights learned by the dense model did not allow any immediate interpretation of importance nor selection of a small set of critical discriminative features.

5 Relationship to Kernel Methods

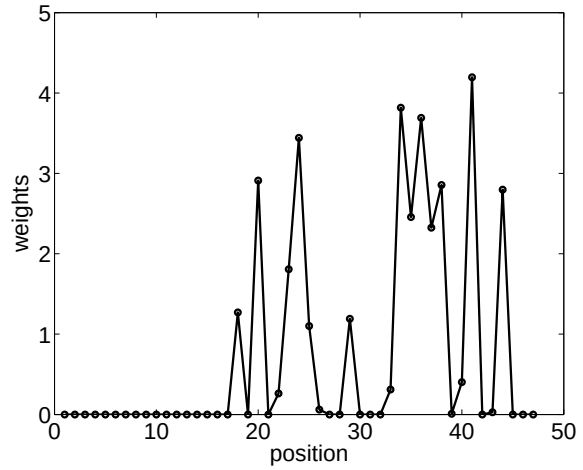
A class of kernels, called *marginalized kernels*, for biological sequences was proposed by Tsuda et al. (2002). Let $x \in \mathcal{X}$ be a set of *observable* variables and $h \in \mathcal{H}$ be a set of *unobservable* (*hidden*) variables. The authors define $K_z(z, z')$ as the *joint kernel*, where $z = (x, h)$. Then in the *marginalized kernel* setting, the similarity between two examples x and x' is defined as:

$$K(x, x') = \sum_{h \in \mathcal{H}} \sum_{h' \in \mathcal{H}} p(h|x) p(h'|x') K_z(z, z'), \tag{15}$$

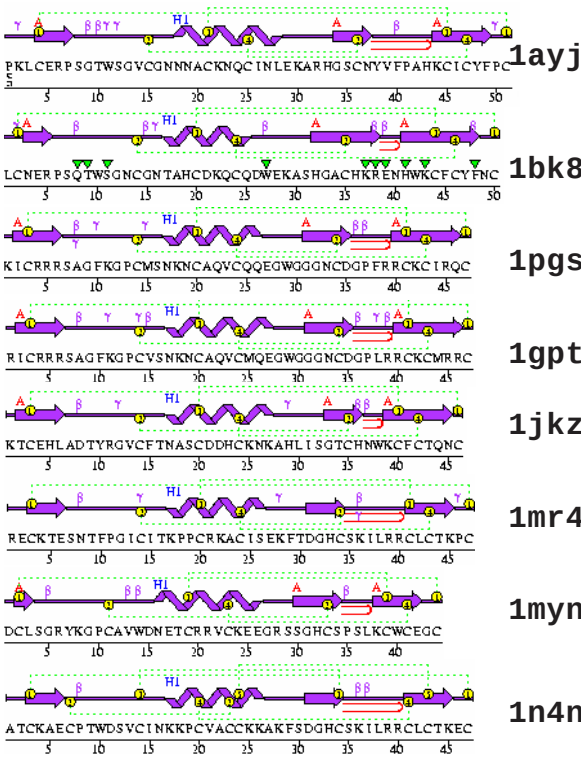
where we sum over all hidden variables. Further, given two sequences X and Y, define the *count kernel* as:

$$K(X, Y) = \sum_{\sigma \in \Sigma} c_\sigma(X) c_\sigma(Y), \tag{16}$$

where $c_\sigma(X)$ denotes the number of times the symbol σ occurs in sequence X. Finally, let $\mathcal{H}$ be the set of states in an HMM, Tseuda *et al.* showed that the

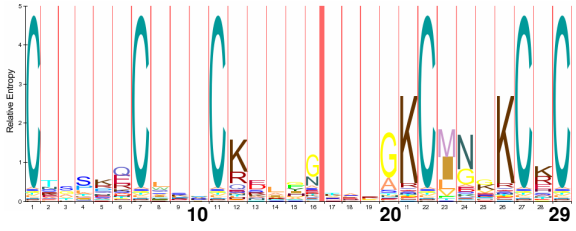


(a)

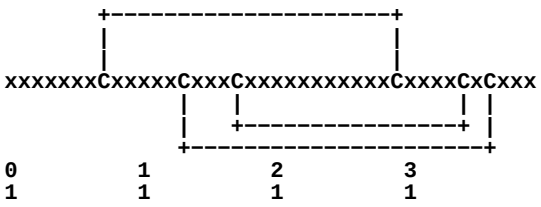


(b)

Figure 6 Panel (a): The sum of the positive coefficients in each position for the *Plant defensins* family. Panel (b): The primary and secondary structure, in schematic diagrams, of eight sequences belonging to the this family. We obtain the diagrams from PDBsum.



(a)



(b)

Figure 7 Panel (a): The HMM-logo of the *short-chain scorpion toxins* family. This logo is obtained from PFam. Panel (b): the schematic representation of this family suggested by PFam.

corresponding *marginalized count kernel* is:

$$\begin{aligned} K(X,Y) &= \sum_{h,h'} p(h|X)p(h'|Y)K_z(Z_X, Z_Y) \\ &= \sum_{\tilde{x},\tilde{s}} \xi(\tilde{x},\tilde{s}|X)\xi(\tilde{x},\tilde{s}|Y) \end{aligned} \tag{17}$$

for all $\tilde{x} \in \Sigma$ and $\tilde{s} \in S$, where S represents the set of states in the HMM. Eq. (17) indicates that, in a kernel setting, the kernel induced by our feature set is a *marginalized count kernel*. It should also be noted, that Eq. (16) implies that the *spectrum-k* kernel is a k-th order count kernel, in which σ spans through all possible *k-mers* induced by Σ. Finally, Tsuda et al. (2002) showed that the *SVM-Fisher* kernel is a special case of marginalized count kernel.

5.1 Comparison with the Spectrum-k kernel

In Eq. (17), each hidden variable is associated with a posterior probability obtained from the forward-backward algorithm. We call these the *soft labels*. However, one may also use *hard labels* in such setting: determine the labels using the Viterbi sequence, the most probable path (Rabiner (1990)) that generated the observed sequence. Use of the Viterbi sequence results in the following similarity measure between two sequences X and Y:

$$\begin{aligned} K_V(X,Y) &= \sum_{t,t'} \sum_{\tilde{x}} [I(X_t = Y_{t'} = \tilde{x})] \cdot \\ &\quad [\sum_{\tilde{s}} I(V(t|X) = V(t'|Y) = \tilde{s})], \end{aligned} \tag{18}$$

where $V(t|X)$ denotes the state that symbol X_t aligns to in the Viterbi sequence; on the other hand, the similarity between X and Y defined by the *Spectrum-k* kernel with $k = 1$ is:

$$K_{S(1)}(X,Y) = \sum_{t,t'} \sum_{\tilde{x}} I(X_t = Y_{t'} = \tilde{x}). \tag{19}$$

As a result, the kernel induced by our feature set also has a close relationship with the Spectrum kernel. The extra term in the end of Eq. (18) is imposed by our feature extractor, in this case, the profile HMM. The impact of this term can be seen in the following example. Consider a new sequence Y', obtained by randomly permuting Y; then it is very likely that $K_V(X,Y') \neq K_V(X,Y)$ because Y' will align differently with the feature extractor; on the other hand, it is clear that $K_{S(1)}(X,Y') = K_{S(1)}(X,Y)$. We believe that this is one of the reasons that in the string kernel setting, k must be some moderately large number, such as 5. In contrast, our equivalent kernel is able to exploit the existence of latent match states in computing a tractable and empirically effective similarity score.

Furthermore, upon close examination of Eq. (18) and Eq. (19), the kernel induced by our features, using the Viterbi path, *is a Spectrum-1* kernel with an *augmented alphabet set*. While the *Spectrum-k* kernel uses the 20 amino acids as the alphabet set, the kernel induced by our features augments the alphabet set, Σ, to Σ', where $\Sigma' = \Sigma \times Z_m^+$, where Z_m^+ is the set of all positive integers up to m, the number of *match* states in the profile HMM.

As indicated in Sec. 2, for *mismatch(k,m_k)* kernel, computing each element in the kernel matrix requires $O(k^{m_k+1}|\Sigma|^{m_k}(T_X + T_Y))$ time. Denote m as the number of *match* states in the profile HMM; using our *hybrid* model with a linear kernel, computation of each element in the kernel matrix requires $O(m)$ time, since only the inner product of the sufficient statistics of two sequences needs to be computed. The complexity of the forward-backward procedure, required to obtain the sufficient statistics of a sequence X with length T_X is $O(mT_X)$; the complexity is *linear*, instead of quadratic, in m, because of the *linear* structure of a profile HMM. Finally, the complexity of constructing the feature extractor (profile HMM) is $O(mnT)$, where n is the number of *labeled, positive* training sequences and T is the length of the longest sequence in the training set. Among 54 experiments, the total number of positive sequences is 1398; each family on average has 26 positive training sequences. Each profile HMM on average has 123 *match* states; the average positive sequence length is 147 residues per sequence. Each profile on average takes 12 E-M like iterations to train and takes 105 seconds on a $2.80GHz$(x2) machine with 1024MB of RAM; as for logistic models, given the features, among 54 families, it takes BBR on average 1 minute to estimate a model and 3 seconds to predict classification results per experiment.

More recent methods based on *profile kernels* (Kuang et al. (2004)) have shown significant promise. Unlike our setting, kernel profile methods leverage the benefits of unlabeled data. Also, each sequence is represented by a profile, resulting in increased computational complexity of the classification approach. As a result, the method is fundamentally different from those compared in this paper and we do not include the comparison in the current study.

6 Conclusion

In this paper we introduce a method for learning *sparse* feature models for the homology detection problem. To extract the features, we use a profile HMM that represents the family of interest. These features are the sufficient statistics of the query sequence with respect to the designed HMM profile. As such, the features offer insight to the underlying evolutionary process such as the degree of conservation of each position in the superfamily.

Using interpretable logistic classifiers with Laplace priors, the learned models exhibit more than 90% reduction in the number of selected features. These results indicate that it may be possible to discover very *sparse* models for certain protein superfamilies, which might confirm the hypotheses suggested in Kister et al. (2002); Reva et al. (2002); Kister et al. (2001) that a small subset of positions and residues in protein sequences may be sufficient to discriminate among different protein classes. We show that the *sparse* model select some *critical positions* that are consistent with current reports. However, at present, the full set of selected positions may not fully agree with the proposed hypotheses. Further analysis is needed to study the correspondences between the computation and hypothesized models.

In our future work we will further investigate and consider biological interpretation of the resulting *sparse* models. In addition, we will expand our framework to utilize additional sets of physically motivated features as well as the unlabeled data, leveraging the benefits of large training sets.

Acknowledgements

This work was done through graduate student support through DIMACS under NSF grants CCF-04-32013 and EIA-02-05116 to Rutgers University.

References

Bairoch, A., Apweiler, R., Wu, C. H., Barker, W. C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M. J., Natale, D. A., O'Donovan, C., Redaschi, N., and Yeh, L.-S. L. (2005). The Universal Protein Resource (UniProt). *Nucl. Acids Res.*, 33(suppl-1):D154–159.

Bateman, A., Coin, L., Durbin, R., Finn, R. D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E. L. L., Studholme, D. J., Yeats, C., and Eddy, S. R. (2004). The Pfam protein families database. *Nucleic Acids Research*, 32(Database-Issue):138–141.

Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., and Wheeler, D. L. (2005). Genbank. *Nucl. Acids Res.*, 33(suppl-1):D34–38.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The Protein Data Bank. *Nucleic Acids Research*, 28:235–242.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Jorunal of the Royal Statistical Society. Series B*, 1977(1):1–38.

Eddy, S. (1998). Profile hidden Markov models. *Bioinformatics*, 14(9):755–763.

Genkin, A., Lewis, D. D., and Madigan, D. (2006 (to appear)). Large-Scale Bayesian Logistic Regression for Text Categorization.

Gribskov, M., McLachlan, A., and Eisenberg, D. (1987). Profile analysis: detection of distantly related proteins. *Proceedings of the National Academy of Sciences*, 84:4355–4358.

Gribskov, M. and Robinson, N. (1996). Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching.

Henikoff, S. and Henikoff, J. G. (1994). Position-based sequence weights. *J Mol Biol.*, 243(4):574–8.

Hulo, N., Bairoch, A., Bulliard, V., Cerutti, L., De Castro, E., Langendijk-Genevaux, P. S., Pagni, M., and Sigrist, C. J. A. (2006). The PROSITE database. *Nucl. Acids Res.*, 34:D227–230.

Jaakkola, T., Diekhans, M., and Haussler, D. (1999). Using the Fisher kernel method to detect remote protein homologies. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 149–158. AAAI Press.

Jaakkola, T., Diekhans, M., and Haussler, D. (2000). A discriminative framework for detecting remote protein homologies. In *Journal of Computational Biology*, volume 7, pages 95–114.

Kister, A. E., Finkelstein, A. V., and Gelfand, I. M. (2002). Common features in structures and sequences of sandwich-like proteins. *PNAS*, 99(22):14137–14141.

Kister, A. E., Roytberg, M. A., Chothia, C., Vasiliev, J. M., and Gelfand, I. M. (2001). The sequence determinants of cadherin molecules. *Protein Sci*, 10(9):1801–1810.

Kuang, R., Ie, E., Wang, K., Wang, K., Siddiqi, M., Freund, Y., and Leslie, C. S. (2004). Profile-based string kernels for remote homology detection and motif extraction. In *CSB*, pages 152–160.

Laskowski, R. A., Chistyakov, V. V., and Thornton, J. M. (2005). PDBsum more: new summaries and analyses of the known 3D structures of proteins and nucleic acids. *Nucl. Acids Res.*, 33:D266–268.

Leslie, C. S., Eskin, E., and Noble, W. S. (2002a). The spectrum kernel: A string kernel for svm protein classification. In *Pacific Symposium on Biocomputing*, pages 566–575.

Leslie, C. S., Eskin, E., Weston, J., and Noble, W. S. (2002b). Mismatch string kernels for svm protein classification. In *NIPS*, pages 1417–1424.

Liao, L. and Noble, W. S. (2002). Combining pairwise sequence similarity and support vector machines for remote protein homology detection. In *RECOMB*, pages 225–232.

Lo Conte, L., Ailey, B., Hubbard, T., Brenner, S., Murzin, A., and Chothia, C. (2000). SCOP: a structural classification of proteins database. *Nucleic Acids Res.*, 28:257–259.

Rabiner, L. R. (1990). A tutorial on hidden Markov models and selected apllications in speech recognition. In Waibel, A. and Lee, K.-F., editors, *Readings in Speech Recognition*, pages 267–296. Kaufmann, San Mateo, CA.

Reva, B., Kister, A., Topiol, S., and Gelfand, I. (2002). Determining the roles of different chain fragments in recognition of immunoglobulin fold. *Protein Eng.*, 15(1):13–19.

Sjolander, K., Karplus, K., Brown, M., Hughey, R., Krogh, A., Mian, I., and Haussler, D. (1996). Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. *Comput. Appl. Biosci.*, 12(4):327–345.

Altschul, S., Gish, W., Miller, W., Myers, E., and Lipman, D. (1990). Basic Local Alignment Search Tool. *Journal of Molecular Biology*, pages 403–410.

Pearson, W. and Lipman, D. (1988). Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences of the United States of America*, 85:2444–2448.

Tsuda, K., Kin, T., and Asai, K. (2002). Marginalized kernels for biological sequences. *Bioinformatics*, 18(suppl_1):S268–275.