Profile-based string kernels for remote homology detection and motif extraction

Rui Kuang[1], Eugene Ie[1,3], Ke Wang[1], Kai Wang[2], Mahira Siddiqi[2],
Yoav Freund[1,3,4], Christina Leslie[1,3,4]*
[1]Department of Computer Science, [2]Department of Biomedical Informatics,
[3]Center for Computational Learning Systems
[4]Center for Computational Biology and Bioinformatics
Columbia University

Abstract

We introduce novel profile-based string kernels for use with support vector machines (SVMs) for the problems of protein classification and remote homology detection. These kernels use probabilistic profiles, such as those produced by the PSI-BLAST algorithm, to define position-dependent mutation neighborhoods along protein sequences for inexact matching of k-length subsequences ("k-mers") in the data. By use of an efficient data structure, the kernels are fast to compute once the profiles have been obtained. For example, the time needed to run PSI-BLAST in order to build the profiles is significantly longer than both the kernel computation time and the SVM training time. We present remote homology detection experiments based on the SCOP database where we show that profile-based string kernels used with SVM classifiers strongly outperform all recently presented supervised SVM methods. We also show how we can use the learned SVM classifier to extract "discriminative sequence motifs" – short regions of the original profile that contribute almost all the weight of the SVM classification score – and show that these discriminative motifs correspond to meaningful structural features in the protein data. The use of PSI-BLAST profiles can be seen as a semi-supervised learning technique, since PSI-BLAST leverages unlabeled data from a large sequence database to build more informative profiles. Recently presented "cluster kernels" give general semi-supervised methods for improving SVM protein classification performance. We show that our profile kernel results are comparable to cluster kernels while providing much better scalability to large datasets.

Keywords: protein classification, support vector machine, kernels, protein motifs.

Supplementary website:
http://www.cs.columbia.edu/compbio/profile-kernel.

1. Introduction

There has been much recent work on support vector machine (SVM) [4] approaches for the classification of protein sequences into functional and structural families and for remote homology detection. Most of this research effort focuses on finding useful representations of protein sequence data for SVM training, either using explicit feature vector representations or *kernel* functions – specialized sequence similarity functions that define an inner product in an implicit feature space for the SVM optimization problem. Among the approaches that have been presented are the Fisher-SVM method [12], which represents each protein sequence as a vector of Fisher scores extracted from a profile hidden Markov model (HMM) for a protein family, and kernels that extend the Fisher kernel method [20]; families of efficient string kernels [16, 15, 17], such as the mismatch kernel, which are based on inexact-matching occurrences of k-length subsequences ("k-mers"); the SVM-pairwise approach [18], which uses a feature vector of pairwise alignment scores between the input sequence and a set of training sequences; the eMOTIF kernel [3], where the feature vector represents counts of occurrences of eMOTIF patterns in the sequence; and feature vectors defined by structure-based I-sites motifs [9]. These studies show that most of the methods achieve comparable classification performance on benchmark datasets, though there are significant differences in computational efficiency [15]. Interestingly, except for the Fisher kernel method and its extensions, these representations do not make intrinsic use of standard tools for protein sequence analysis such as profiles [7] and profile HMMs [14, 6, 2] – more commonly, they use scores based on alignment or probabilistic models to construct a large set of features. It is perhaps surprising that very general k-mer based string kernels perform as well as the Fisher kernel ap-

* Corresponding author. Mailing address: 1214 Amsterdam Ave, MC 0401, New York, NY 10027. Email: `cleslie@cs.columbia.edu`. Telephone: 1-212-939-7043. Fax: 1-212-666-0140

proach, which makes well-motivated use of profile HMMs [15].

In this paper, we define a natural extension of the k-mer based string kernel framework to define kernels on protein sequence profiles, such as those produced by PSI-BLAST [1]. We choose to use profiles (rather than more complex models) because they can be calculated by PSI-BLAST in a tractable amount of time and because, once the profiles are obtained, we can efficiently compute string kernel values using an appropriate data structure; in fact, the time needed to compute the profile kernel matrix and the SVM training time are significantly shorter than the time needed by PSI-BLAST to compute profiles. From a machine learning point of view, use of PSI-BLAST profiles can be viewed as a *semi-supervised* approach – that is, a method that learns both from labeled training examples (sequences whose structural classification is known) and unlabeled examples – an important consideration given the relatively small amount of labeled data in this problem. Through iterative heuristic alignment, PSI-BLAST leverages unlabeled data from a large sequence database to obtain a much richer profile representation of each sequence. Intuitively, this richer data representation, made available to an SVM through a profile-based kernel, should greatly improve classification performance. Also, profile-based kernels are a significantly different semi-supervised approach than the Fisher-SVM method: with the Fisher kernel, unlabeled data in the form of domain homologs are used to train a model for a protein family of sequences in the training set, and then each sequence is represented by sufficient statistics with respect to the learned model; in our approach, unlabeled data is used to produce a profile model for each training sequence independently, and then the kernel is defined on the profiles. Our experimental results for the remote homology detection task, using a benchmark based on the SCOP database, show that our profile-based kernel used with SVM classifiers strongly outperform all the recent purely supervised SVM methods that we compared against.

Usually, SVM methods are treated as a "black box" method, since in general it is difficult to interpret the SVM classification rule. For the case of profile string kernels, we show how we can use the trained SVM classifiers to define positional scores along the protein profiles that define a smoothed contribution to the positive classification decision. We find that on average just over 10% of the profile contributes 90% of the total score for positive training sequences, and thus we can extract distinguished regions that we call "discriminative sequence motifs". We give examples from our SCOP dataset to show that these discriminative motifs correspond to meaningful structural features, giving a proof of principle that the SVM-profile kernel approach allows us to extract useful sequence information.

Recently presented "cluster kernels" approaches [21] give general semi-supervised methods for improving SVM protein classification performance of a base kernel using unlabeled data together with a similarity measure on input examples. These cluster kernels were successfully applied to the protein classification problem using the mismatch kernel as a base kernel for sequence data and BLAST or PSI-BLAST to define similarity scores. However, for large amounts of unlabeled data, these more general methods do not scale as well as our profile kernel approach. We show that our profile kernel results are comparable to cluster kernels while providing much better scalability to large datasets.

2. The Profile Kernel

A key feature of the SVM optimization problem is that it depends only on the inner products of the feature vectors representing the input data, allowing us to use *kernel techniques*. If we define a feature map Φ from the input space of protein sequences into a (possibly high-dimensional) vector space called the *feature space*, we obtain a *string kernel* – that is, a kernel on sequence data – defined by $K(x, y) = \langle\Phi(x), \Phi(y)\rangle$.

We first show how to define a feature mapping for protein sequence profiles – more precisely, we consider input examples to be profiles $P(x)$, where x is a sequence $x = x_1x_2 \dots x_N$ from the alphabet Σ of amino acids ($|\Sigma| = 20$, and the length $N = |x|$ depends on the sequence), and $P(x) = \{p_i(a), a \in \Sigma\}_{i=1}^{N}$ is a profile for sequence x, with $p_i(a)$ denoting the emission probability of amino acid a in position i and $\sum_{a\in\Sigma} p_i(a) = 1$ for each position i. We then show how to efficiently and directly compute the profile-based string kernel values $K(P(x), P(y))$ without storing the feature vector representation.

2.1. Profile-defined Mapping to k-mer Feature Space

Following the framework of k-mer based string kernels [16, 15, 17], our profile-based kernels will depend on a feature mapping to the $|\Sigma|^k$-dimensional feature space indexed by the set of all possible k-length subsequences ("k-mers") of amino acids, where k is a small positive integer.

Previous string kernels relied on defining an inexact-matching neigborhood of k-mers around each k-length contiguous subsequence in the input sequence. For example, for the (k, m)-mismatch kernel, one defines the "mismatch neighborhood" around k-mer $\alpha = a_1a_2 \dots a_k$ to be the set of all k-length sequences β from Σ that differ from α by at most m mismatches. For a k-mer α, the mismatch feature map is defined as

$$\Phi^{\text{Mismatch}}_{(k,m)}(\alpha) = (\phi_\beta(\alpha))_{\beta\in\Sigma^k}$$

where $\phi_\beta(\alpha) = 1$ if β belongs to $N_{(k,m)}(\alpha)$, and $\phi_\beta(\alpha) = 0$ otherwise, and one extends additively to full-length sequences x by summing the feature vectors for all the k-mers in x:

$$\Phi^{\text{Mismatch}}_{(k,m)}(x) = \sum_{k\text{-mers } \alpha \text{ in } x} \Phi^{\text{Mismatch}}_{(k,m)}(\alpha)$$

Thus each coordinate of the feature map is a count of the inexact-matching occurrences of a particular k-mer, where mismatching is used to define inexact matching.

For the profile kernel, we use the probabilistic profile $P(x)$ to define a mutation neighborhood for each k-length segment in the input sequence x. Therefore, unlike previous string kernels, the inexact-matching neighborhood k-mers are not the same for all the data but instead vary from sequence to sequence and within different regions of the same sequence. For each k-length contiguous subsequence $x[j+1:j+k] = x_{j+1}x_{j+2}\ldots x_{j+k}$ in x ($0 \le j \le |x|-k$), the *positional mutation neighborhood* is defined by the corresponding segment of the profile $P(x)$:

$$M_{(k,\sigma)}(P(x[j+1:j+k])) =$$
$$\{\beta = b_1 b_2 \ldots b_k : -\sum_{i=1}^{k} \log p_{j+i}(b_i) < \sigma\}.$$

Note that the emission probabilities $p_{j+i}(b), i = 1 \ldots k$, come from the profile $P(x)$ – for notational simplicity, we do not explicitly indicate the dependence on x. Typically, the profiles are estimated from close homologs found in a large sequence database and may be too restrictive for our purposes. Therefore, we smooth the estimates using background frequencies $q(b), b \in \Sigma$, of amino acids in the training dataset via

$$\tilde{p}_i(b) = \frac{p_i(b) + Cq(b)}{1+C}, i = 1 \ldots |x|,$$

where C is a smoothing parameter, and we use the smoothed emission probabilities $\tilde{p}_i(b)$ in place of $p_i(b)$ in defining the mutation neighborhoods.

We now define the profile feature mapping as

$$\Phi^{\text{Profile}}_{(k,\sigma)}(P(x)) =$$
$$\sum_{j=0\ldots|x|-k} (\phi_\beta(P(x[j+1:j+k])))_{\beta\in\Sigma^k}$$

where the coordinate $\phi_\beta(P(x[j+1:j+k])) = 1$ if β belongs to the mutation neighborhood $M_{(k,\sigma)}(P(x[j+1:j+k]))$, and otherwise the coordinate is 0.

The profile kernel is simply defined by the inner product of feature vectors:

$$K^{\text{Profile}}_{(k,\sigma)}(P(x), P(y)) =$$
$$\langle \Phi^{\text{Profile}}_{(k,\sigma)}(P(x)), \Phi^{\text{Profile}}_{(k,\sigma)}(P(y))\rangle.$$

2.2. Efficient Computation of the Kernel Matrix

Rather than storing sparse feature vectors in high-dimensional k-mer space, we directly and efficiently compute the kernel matrix using a *trie* data structure, similar to the mismatch tree approach previously presented in [16, 15, 17]. The difference for the profile kernels is that, instead of matching k-mers along the path to a leaf, we pass k-length profiles down the tree branches.

Our new (k, σ)-profile trie is a rooted tree of depth k where each internal node has $|\Sigma| = 20$ branches, each labeled with an amino acid (symbol from Σ). A leaf node still represents a fixed k-mer in our feature space, obtained by concatenating the branch symbols along the path from root to leaf. We perform a depth-first traversal of the data structure and store, at a node of depth d, a set of pointers to all k-length profiles $P(x[j+1 : j+k])$ from the sample data set, whose current cumulative substitution scores, up to depth d, are less than the σ threshold, that is, $-\sum_{i=1}^{d} \log p_{j+i}(b_i) < \sigma$, where $b_1 ... b_d$ is the prefix of the current node. As we pass from a parent node at depth d to a child node at depth d+1 along a branch with symbol label b, we add for each k-length profile $P(x[j+1:j+k])$ a score $-\log p_{j+d+1}(b)$. Only those profile segments whose cumulative substitution scores are still less than σ will be passed to the child node. At the leaf node, we update the kernel by computing the contribution of active profile segments to the corresponding k-mer feature.

The complexity of computing each value $K(x, y)$ depends on the size of the positional mutation neighborhood of k-length profiles. With a typical choice of σ, we empirically observe that the mutation neighborhood allows about $m = 1$ or 2 mismatches relative to the original k-mer for the k-length profile in sequence x. Thus we can estimate that the running time is bounded by that of (k, m)-mismatch kernel, which is $O(k^{m+1}|\Sigma|^m(|x| + |y|))$, with $m \le 2$. See Section 3 for actual running times in benchmark experiments.

2.3. Extraction of Discriminative Motifs

Using the PSI-BLAST sequence profiles and the learned SVM weights, we can do a positional analysis to determine which regions of the positive training sequence contribute most to the classification score and thus extract "discriminative" protein motif regions. For a training set of protein sequence $\{x_i\}_{i=1}^n$, the normal vector to the SVM decision hyperplane is given by:

$$\mathbf{w} = \sum_{i=1}^{n} y_i c_i \Phi^{\text{Profile}}_{(k,\sigma)}(P(x_i)),$$

where the c_i are learned weights and $y_i \in \{\pm 1\}$ are training labels. For each k-length profile segment of sequence x, its contribution to the classification score is (up to a constant):

$$S(x[j+1:j+k]) = \\ \langle \phi^{\text{Profile}}_{(k,\sigma)}(P(x[j+1:j+k])), \mathbf{w}\rangle.$$

We are mainly interested in discriminative motifs that contribute to the positive decision of the classifier, so we define a positional score for each position j in a (positive) training sequence by summing up positive contributions of k-length segments containing the position:

$$\sigma(x[j]) = \sum_{q=1}^{k} \max(S(x[j-k+q:j-1+q]), 0).$$

We now sort these positional scores (for all positions in all positive training sequences) in decreasing order $\sigma(x[j_1]) \geq \sigma(x[j_2]) \geq \ldots \geq \sigma(x[j_N])$, and we find the first index M such that cumulative sum $\sum_{i=1}^{M} \sigma(x[j_i])$ is greater than .9 times the total sum $\sum_{i=1}^{M} \sigma(x[j_i])$. Thus positions $j_1, \ldots j_M$ constitute 90% of the positionally averaged positive classification scores. We will see in the Section 3.2 that these positions tend to fall in short segments of the protein sequence; we call these segments "discriminative motif regions".

3. Experiments

We test SVM classification performance of profile-based string kernels against other recently presented SVM methods on a SCOP benchmark dataset. Methods are evaluated on the ability to detect members of a target SCOP family (positive test set) belonging to the same SCOP superfamily as the positive training sequences; no members of the target family are available during training. We use the same experimental set-up that has been used in several previous studies of remote homology detection algorithms [12, 18].

We use the same 54 target families and the same test and training set splits as in the remote homology experiments in [18]. The sequences are 7329 SCOP domains obtained from version 1.59 of the database after purging with astral.stanford.edu so that no pair of sequences share more than 95% identity. Compared to [18], we reduce the number of available labeled training patterns by roughly a third. Data set sequences that were neither in the training nor test sets for experiments from [18] are considered to be additional unlabeled data, used for cluster kernel method we compare against. All methods are evaluated using the receiver operating characteristic (ROC) score and the ROC-50, which is the ROC score computed only up to the first 50 false positives [8].

We computed the profiles needed for our kernels by running PSI-BLAST [1] from the nonredundant database with default search parameters and with background frequencies, used for smoothing, estimated from the full dataset of 7329 SCOP domains. However, we limited the maximum number of iterative database searches to 2 iterations in order to speed up PSI-BLAST computation. We used smoothing parameter corresponding to $\frac{1}{1+C} = .8$ in the profile kernel computation. The time needed to compute PSI-BLAST profiles for all sequences was approximately 36 hours on a 2.2 GHz Linux server (using at most 2 iterative database searches); on the same CPU, the time required to compute the 7329 x 7329 kernel matrix was 10 hours, and all 54 SVM experiments were completed in 30 minutes.

3.1. SCOP Experiments: Comparison with Supervised and Semi-Supervised Methods

We compared the results of profile kernels with three recently presented SVM methods, using different representations of protein sequence data – the eMOTIF kernel, the SVM-pairwise method, and the mismatch kernel – as well as recent semi-supervised cluster kernel methods [21]. We also compared the SVM methods to PSI-BLAST, used directly as a method for ranking test sequences relative to positive training sequence queries (see below).

We used the eMOTIF database extracted from eBlocks and packaged with eBAS version 3.7 [10, 19], and we obtained code for computing eMOTIF feature vectors from the authors [3]. For the the SVM-pairwise method, we used PSI-BLAST E-values as pairwise similarity scores (see [21] for details on this representation). We note that this use of PSI-BLAST with the SVM-pairwise method is not fully-supervised, because the PSI-BLAST scores themselves make use of unlabeled data. For the mismatch kernel, we use $(k, m) = (5, 1)$ as presented in the original paper [16].

We include results for PSI-BLAST, used directly as a ranking method, in order to provide a baseline comparison with a widely used remote homology detection method and also to demonstrate the added benefit of combining PSI-BLAST with our SVM string kernel approach. The PSI-BLAST algorithm, which iteratively builds up a probabilistic profile for a query sequence by searching a large database, also gives a faster approximation to the iterative training method of profile HMMS. (We do not test profile HMMs here due to computational expense, but in previous benchmark results for the remote homology problem, SVM string kernel and Fisher kernel methods were both found to outperform profile HMMs [16, 12].) Since PSI-BLAST is not a family-based method, we report results by averaging over queries: for each experiment, we use PSI-BLAST with each of the positive training sequences as the query and search against the non-redundant database in order to produce a set of profiles, and then we use these profiles to rank the test set sequences by their PSI-BLAST E-values. The ROC (ROC-50) score that we report for the experiment is the average of all ROC (ROC-50) scores from these rankings. (We note that a more sophisticated PSI-BLAST training procedures that uses all positive training sequences at once might be possible, but it is not clear how best to do this given the diverse positive training set.) For the PSI-BLAST ranking method, we use PSI-BLAST with the default parameters, allowing a maximum of 10 iterative searches against the nonredundant protein database in order to build the profiles.

In our main experiments, we computed the profile kernel with PSI-BLAST profiles built using at most 2 iterative searches in order to reduce the PSI-BLAST computation cost. Therefore, the profiles used in our kernels were somewhat less accurate than those used for the PSI-BLAST ranking method. We tested profile kernels with $(k, \sigma) = (4, 6.0), (5, 7.5)$ and $(6, 9.0)$; these parameter choices all yield similar results. Figure 1 shows the comparison of SVM performance of the $(5, 7.5)$-profile kernel against the PSI-BLAST ranking method, the eMOTIF kernel, the mismatch kernel, and the SVM-pairwise method using PSI-BLAST across the 54 experiments in the benchmark. A signed rank test with Bonferroni correction for multiple comparisons concludes that the profile kernel significantly outperforms the mismatch kernel (p-value 1.7e-06), SVM-pairwise kernel (8.4e-04), eMOTIF kernel (2.2e-07), and mean PSI-BLAST ranking (2.6e-08). Average ROC and ROC-50 scores across the experiments for all methods are reported in Table 1.

To show the effect of using more accurate PSI-BLAST profiles, we also include in Table 1 profile kernel results based on PSI-BLAST profiles that were trained for up to 5 search iterations. The results demonstrate that we can have significant improvement in ROC and ROC-50 scores for the profile kernel method by improving the profiles. However, running PSI-BLAST for 5 iterations instead of 2 iterations increased the PSI-BLAST computation time by an order of magnitude. Therefore, in our subsequent analysis, we refer only to the first set of SVM classifiers, which use profiles based on up to 2 PSI-BLAST iterations.

We note that the original authors of the SVM-pairwise used Smith-Waterman scores (SW) for pairwise comparison scores; however, on a similar benchmark with more training data than the current dataset, results for SVM-pairwise with SW scores were weaker than the PSI-BLAST results reported here, and ROC performance was only slightly better (ROC = 0.893, ROC-50 = 0.434). Thus the semi-supervised PSI-BLAST scores do indeed give a richer and more effective representation for SVM-pairwise; however, using PSI-BLAST profiles to define a profile-based string kernel is clearly more effective than SVM-pairwise with PSI-BLAST.

Our SCOP dataset is different from and larger than the benchmark on which the eMOTIF kernel was originally tested [3]. In cases where a superfamily has a common eMOTIF pattern or set of patterns, the eMOTIF kernel should achieve good specificity. We speculate that in our 54 experiments, fewer superfamilies are characterized by common eMOTIF patterns and that accordingly the eMOTIF kernel achieves weaker performance.

We do not include a comparison against the Fisher kernel method [12], again due to computational expense, but based on previous comparisons, we expect performance to be similar to that of the mismatch kernel [16].

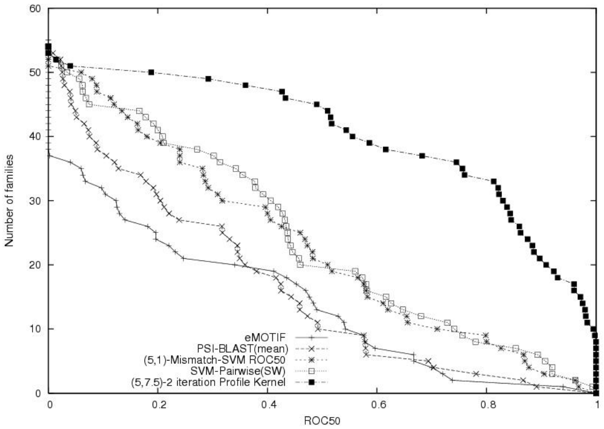


Figure 1: **Comparison of recent SVM-based homology detection methods for the SCOP 1.59 benchmark dataset.** The graph plots the total number of families for which a given method exceeds an ROC-50 score threshold. Each series corresponds to one of the homology detection methods described in the text.

Finally, we also compared our profile kernel against recently presented cluster kernels methods [21]. These methods use "clustering" of additional unlabeled sequence data to improve the base representation. Here, sequences from the original SCOP dataset of 7329 domains that are not used in the training or test sets of any experiment provide the unlabeled data. For simplicity, we give results for only one of the two novel cluster kernel methods from [21], the neighborhood kernel. (Results for the bagged kernel are very similar but more time-consuming to compute.) The neighborhood kernel uses the $(5, 1)$-mismatch kernel as the base kernel and uses PSI-BLAST to define "neighborhood sets" $\text{Nbd}(x)$ around each input sequence x, consisting of labeled or unlabeled sequences x' with similarity score to x below E-value threshold of .05, together with x itself. Then the implicit feature vector is $\Phi_{nbd}(x) = \frac{1}{|\text{Nbd}(x)|} \sum_{x' \in \text{Nbd}(x)} \Phi_{\text{Mismatch}}(x')$.

We see from figure 2 that the profile kernel has similar

Kernel	ROC	ROC-50
eMOTIF	0.711	0.247
PSI-BLAST(mean)	0.7429	0.2925
Mismatch(5,1)	0.870	0.416
SVM-pairwise(PSI-BLAST)	0.866	0.533
Neighborhood	0.923	0.699
Profile(4,6.0)-2 iterations	0.939	0.700
Profile(5,7.5)-2 iterations	0.945	0.735
Profile(6,9.0)-2 iterations	0.952	0.731
Profile(4,6.0)-5 iterations	0.955	0.776
Profile(5,7.5)-5 iterations	0.959	0.782
Profile(6,9.0)-5 iterations	0.967	0.784

Table 1: **Mean ROC and ROC-50 scores over 54 target families.**

performance to the neighborhood kernel (the slight preference to profile kernel is not significant by a signed rank test with p-value threshold of 0.05). We note that our profile kernel is making use of more unlabeled data than the neighborhood kernel, since the neighborhoods are based on a smaller unlabeled database. However, as we scale up, computing the neighborhood kernel for extremely large neighborhood sets of sequences becomes expensive (computation time scales linearly with the size of the neighborhood). One can randomly select sequences from the neighborhood, but then one still has to devise an appropriate way of computing a sample without storing many thousands of sequences. (The bagged kernel from [21] has similar scalability issues as the database gets large.) By comparison, the profile-based string kernel approach achieves good SVM performance and computational efficiency while only representing the sequence profiles.

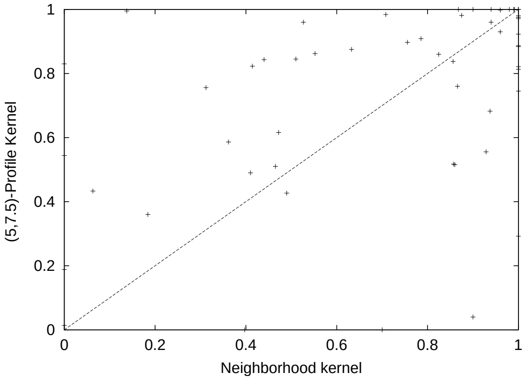


Figure 2: **Comparison of profile kernel (using 2 PSI-BLAST iterations) with recent cluster kernel approaches on the SCOP 1.59 benchmark dataset.** The graph plots ROC-50 scores of the profile kernel (y-axis) versus the neighborhood cluster (x-axis), a recent cluster kernel method, for the 54 experiments in the SCOP benchmark.

3.2. Motif Extraction from SVM Predictions

We next calculated positional contribution scores $\sigma(x[j])$ for our trained SVM classifiers, as outlined in Section 2.3, to analyze which parts of the positive training sequences were most important for positive classification. Typically, we found peaky distributional plots of $\sigma(x[j])$ along positive training sequences, as shown for one experiment in Figure 3: the peaks in these plots correspond to "discriminative motif regions". From cumulative contribution analysis, we found that on average across the 54 experiments, 10.4% of the positions in the positive training sequences gave a cumulative total of 90% of the SVM classification score for these sequences.

We manually examined the motif candidates for positive training sequence sets in 13 experiments (2 sets from all-α class, 5 from all-β class, 5 from α+β class, and 1 from small proteins class) with high ROC scores. By comparing them with PDB annotations, we tried to identify common functional and structural characteristics captured by motif candidates for these superfamilies. We found results of four experiments to be of particular interest. We describe two of these experiments below; results for the other two experiments are available on the supplementary website.

One interesting example came from the homology detection experiment for PH domain-like protein superfamily (SCOP 1.59 superfamily 2.55.1). Proteins in this superfamily share a conserved fold made up of a beta-barrel composed of two roughly perpendicular, anti-parallel beta-sheets and a C-terminal alpha helix. Previous studies have shown that PH domains bind to their inositol phosphate ligands via a binding surface composed primarily of residues from the $\beta1/\beta2$, $\beta3/\beta4$, and $\beta6/\beta7$ loops [11]. The motif candidates we extracted correspond well with the C-terminal alpha helix and the ligand-binding region at the $\beta1/\beta2$, $\beta3/\beta4$, and $\beta6/\beta7$. In Figure 4, we show the motif regions for one member of this superfamily, mouse beta-spectrin protein, together with structural and functional annotations.

A second interesting example was the homology detection experiment for the scorpion toxin-like superfamily (SCOP 1.59 superfamily 7.3.7). By examining the motif candidates from all sequences in this superfamily, we find a common motif region that forms a beta-hairpin with two adjacent disulphides. Previous studies have found that this hairpin structure might be structurally important in interacting with membrane receptors and ionic channels for proteins in this superfamily, and the disulphide bridges can help to stabilize the toxin protein. Figure 5 gives an example from this superfamily, the scorpion OSK1 toxin protein, to demonstrate the structure of the motif candidate [13].

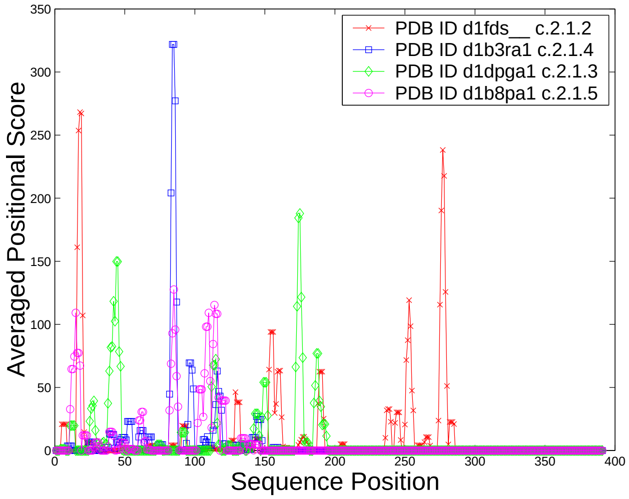
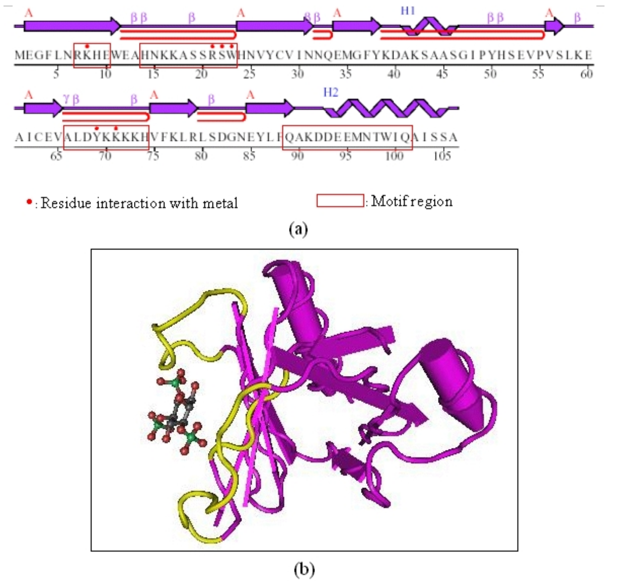


Figure 3: **Positional contribution analysis of SVM classification score for SCOP superfamily 3.2.1 (target family 3.2.1.7).** The plot shows the the contribution of each position along the sequence, obtained by averaging k-mer profile SVM scores for all k-mers containing the position, for positive training sequences in the experiment.

3.3. Discriminative regions versus protein motif databases

To analyze our discriminative motif candidates further, we consider whether the discriminative regions that we found coincide with known protein motifs from the eMOTIF database [3] or structural motifs from the I-sites library[5]. For a simple comparison, we compute the extent to which eMOTIF and I-sites motifs contribute to the overall positive discriminative scores for positive training sequences. We calculate accumulated discriminative scores falling into the sequence regions matched by any motif from the eMOTIF database or the I-sites database, and then we compare it with the expected contribution based on motif coverage, which is estimated by the ratio between total length of motif regions and the sequence length. We also compute the ratio of the eMOTIF/I-sites contribution to the expected contribution.

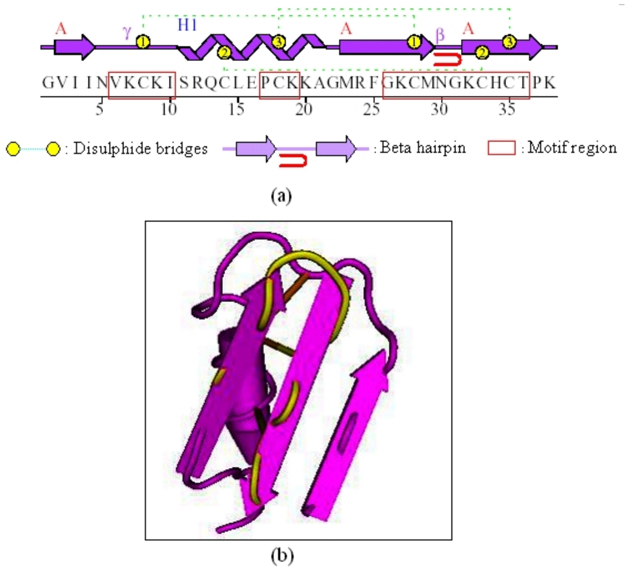
Interestingly, we found that on average, the eMOTIF/I-sites contribution to the discriminative score is slightly, but not dramatically, higher than expected. We show this comparison in Table 2 for eMOTIF and Table 3 for I-sites, giving results for different confidence thresholds using the eBAS and I-sites software, respectively. We conclude that our discriminative motif regions provide information that is complementary or additional to eMOTIF or I-sites motifs in many experiments.


A
β β
β
A
β β
A
H1
β β
A
β
MEGFLNRKHEWEAHNKKASSRSWHNVYCVINNQEMGFYKDAKSAASGIPYHSEVPVSLKE
5
10
15
20
25
30
35
40
45
50
55
60
A
γ β
β
A
β
A
H2
AICEVALDYKKKHVFKLRLSDGNEYLFQAKDDEEMNTWIQAISSA
65
70
75
80
85
90
95
100
105
●: Residue interaction with metal
▭: Motif region

(a)

(b)

Figure 4: **Motif regions on the Mouse beta-spectrin protein that belongs to the PH domain-like protein superfamily.** (a) PDB sequence annotation (PDB id 1btn) and SVM-extracted motif regions. (b) 3D structure of the mouse beta-spectrin showing the SVM-extracted motif regions on the protein structure. The yellow regions are the motif regions; the molecule is shown in pink and the ligand in green.



(a)

(b)

Figure 5: **Motif regions on the scorpion OSK1 Toxin from the Scorpion toxin-like superfamily.** (a) PDB sequence annotation (PDB id 1sco) and SVM-extracted motif regions. (b) 3D of the OSK1 toxin showing the SVM-extracted motif regions on the protein structure. The yellow regions are the motif regions; the orange bars represent the disulphide bridges.

eBAS cutoff threshold	Average eMOTIF contribution	Average expected contribution	Average ratio of eMOTIF over expected
-1	0.4694	0.4263	1.1120
-4	0.3079	0.2739	1.1241
-8	0.2502	0.2188	1.1469
-15	0.1739	0.1500	1.1525

Table 2: **Comparison of eMOTIF motifs versus SVM discriminative scores.**

I-sites confidence threshold	Average I-sites contribution	Average expected contribution	Average ratio of I-sites over expected
0.7	0.5419	0.4894	1.1197
0.8	0.3697	0.3354	1.0995
0.9	0.1656	0.1446	1.1038

Table 3: **Comparison of I-sites motifs versus SVM discriminative scores.**

4. Discussion

We have presented a novel string kernel based on protein sequence profiles, such as those produced by PSI-BLAST. The profile kernel extends the framework of k-mer based string kernels but dramatically improves SVM classification and remote homology detection over these earlier kernels. In our SCOP benchmark experiments, the SVM-profile kernel also outperformed other recently presented SVM approaches such as the eMOTIF kernel and SVM-pairwise and gave far better performance than PSI-BLAST used directly as a ranking method. Furthermore, the profile kernel is competitive with recent semi-supervised cluster kernels, such as the neighborhood kernel, while achieving much better scalability to large datasets. We note that the cluster kernel approaches are general methods that can be used for a variety of applications, while the profile kernel is specialized for protein sequence data; profiles are often computed and stored for other kinds of protein sequence analysis, so profile-based kernels are particularly convenient.

We also show how to compute positional scores along profiles for the positive training sequences and thus extract discriminative sequence motifs. As a proof of principle, we give examples from preliminary analysis where these discriminative regions indeed map to important functional and structural features of the corresponding superfamilies. These discriminative motifs may be of use to structural biologists for improving comparative models. Moreover, we observed that motifs from known protein motif libraries like eMOTIF and I-sites were only slightly overrepresented in our discriminative regions, suggesting that discriminative motifs for structural categories provide information that is complementary or supplementary to known motif databases. Moreover, in cases where the protein classification to be learned is a functional category, such as enzymatic activity, the method could be used to find discrimative sites associated with protein function.

One significant finding from the analysis of our method was that on average across the remote homology experiments, only about 10% of the positions in the positive training sequences gave a cumulative total of 90% of the SVM classification score for these sequences. This result suggests that the multiple alignment of protein domain sequences from a superfamily – which would be used, for example, in a superfamily-based profile HMM approach – might be unnecessary for this problem, since the discriminative information is concentrated in short subregions of the protein sequences. Our profile-based string kernel approach does implicitly use heuristic alignment via PSI-BLAST, but this is only to build a local profile model around each sequence, not to build a model for all the positive sequences at once. We find that local profile information, when combined with an effective profile-based string kernel representation and a powerful classification algorithm, allows us to implement a new and compelling alternative approach to remote homology detection.

Acknowledgments. This work is supported by an Award in Informatics from the PhRMA Foundation, NIH grant LM07276-02, NSF grant ITR-0312706, and NSF grant CCR-0325463. We thank Asa Ben-Hur for providing his eMOTIF kernel code and Chris Bystroff for providing and helping us with the I-sites package.

References

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

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

[3] A. Ben-Hur and D. Brutlag. Remote homology detection: a motif based approach. *Proceedings of the Eleventh International Conference on Intelligent Systems for Molecular Biology*, 2003.

[4] B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Fifth Annual Workshop on Computational Learning Theory*, pages 144–152, Pittsburg, PA, 1992. ACM Press.

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

[6] S. R. Eddy. Multiple alignment using hidden markov models. In *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, pages 114–120. AAAI Press, 1995.

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

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

[9] Y. Hou, W. Hsu, M. L. Lee, and C. Bystroff. Efficient remote homology detection using local structure. *Bioinformatics*, 19(17):2294–2301, 2003.

[10] J. Y. Huang and D. L. Brutlag. The emotif database. *Nucleic Acids Res.*, 29:202–204, 2001.

[11] M. Hyvonen, M. J. Macias, M. Nilges, H. Oschkinat, M. Saraste, and M. Wilmanns. Structure of the binding site for inositol phosphates in a ph domain. *EMBO J*, 14:4676, 1995.

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

[13] V. A. Jaravine, D. E. Nolde, M. J. Reibarkh, Y. V. Korolkova, S. A. Kozlov, K. A. Pluzhnikov, E. V. Grishin, and A. S. Arseniev. Three-dimensional structure of toxin osk1 from orthochirus scrobiculosus scorpion venom. *Biochemistry*, 36(6):1223–32, 1997.

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

[15] C. Leslie, E. Eskin, A. Cohen, J. Weston, and W. S. Noble. Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 2004. To appear.

[16] C. Leslie, E. Eskin, J. Weston, and W. S. Noble. Mismatch string kernels for SVM protein classification. *Advances in Neural Information Processing Systems 15*, pages 1441–1448, 2003.

[17] C. Leslie and R. Kuang. Fast kernels for inexact string matching. *Sixteenth Annual Conference on Learning Theory and Seventh Kernel Workshop*, pages 114–128, 2003.

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

[19] C. G. Nevill-Manning, T. D. Wu, and D. L. Brutlag. Highly specific protein sequence motifs for sequence analysis. *Proceedings of the National Academy of Sciences 95*, pages 5865–5871, 1998.

[20] K. Tsuda, M. Kawanabe, G. Rtsch, S. Sonnenburg, and K. Müller. A new discriminative kernel from probabilistic models. *Neural Computation*, 14:2397–2414, 2002.

[21] J. Weston, C. Leslie, D. Zhou, A. Elisseeff, and W. S. Noble. Cluster kernels for semi-supervised protein classification. *Neural Information Processing Systems 17*, 2003.