Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 05-007

Profile Based Direct Kernels for Remote Homology Detection and Fold Recognition

Huzefa Rangwala and George Karypis

March 31, 2005

Profile Based Direct Kernels for Remote Homology Detection and Fold Recognition*

Huzefa Rangwala and George Karypis

Department of Computer Science & Engineering/Digital Technology Center
University of Minnesota, Minneapolis, MN 55455
Technical Report # 05–007

rangwala@cs.umn.edu, karypis@cs.umn.edu

Last updated on March 21, 2005 at 3:51pm

Abstract

Motivation: *Remote homology detection between protein sequences is a central problem in computational biology. Supervised learning algorithms based on support vector machines are currently the most effective method for remote homology detection. The performance of these methods depends on how the protein sequences are modeled and on the method used to compute the kernel function between them.*

Results: *We introduce new classes of kernel functions that are constructed by directly combining automatically generated sequence profiles with new and existing approaches for determining the similarity between pairs of protein sequences, which employ effective schemes for scoring the aligned profile positions. Experiments with remote homology detection and fold recognition problems show that these kernels are capable of producing results that are substantially better than those produced by all of the existing state-of-the-art SVM-based methods. In addition, the experiments show that these kernels, even when used in the absence of profiles, produce results that are better than those produced by existing non-profile-based schemes.*

1 Introduction

Breakthroughs in large-scale sequencing have led to a surge in the available protein sequence information that has far out-stripped our ability to experimentally characterize their functions. As a result, researchers are increasingly relying on computational techniques to classify these sequences into functional and structural families based on sequence homology.

While satisfactory methods exist to detect homologs with high levels of similarity, accurately detecting homologs at low levels of sequence similarity (remote homology detection) still remains a challenging problem. Over the years, a large number of methods have been developed for homology detection. Some of the early methods were based on pairwise sequence comparisons computed using either optimal dynamic programming-based algorithms [23, 28] or various fast approximations [25, 2]. Better remote homology prediction was later obtained by comparing a protein with a collection of related proteins using methods such as protein family profiles [5], hidden Markov models (HMMs) [15, 3], PSI-BLAST [1], and SAM [14]. These schemes produced models that were generative in the sense that they built a model for a set of related proteins and then checked to see how well this model explained a candidate protein. In recent years, the performance of remote homology detection was greatly improved through the use of methods that explicitly modeled the differences between the various classes (protein families) and built discriminative models. These methods by using both sequences known to belong to a particular class (positive examples) and sequences known to be outside this class (negative examples) are better suited for identifying and capturing the rather weak sequence-level signals of the remote homology detection problem. A number of different methods have been developed that build these discriminative models using support vector machines (SVM) [29] and have been shown to produce results that are in general superior to those produced by either pairwise sequence comparisons or approaches based on generative models provided that there is sufficient data for training [12, 19, 17, 18, 10, 11, 26, 16].

A core component of an SVM is the kernel function. For our purposes, the kernel function measures the similarity between any pair of examples. Different kernels correspond to different notions of similarity and can lead to discriminative functions with different performance. A common approach for deriving a kernel function is to first choose an appropri-

*This work was supported in part by NSF EIA-9986042, ACI-0133464, ACI-0312828, and IIS-0431135; the Digital Technology Center at the University of Minnesota; and by the Army High Performance Computing Research Center (AHPCRC) under the auspices of the Department of the Army, Army Research Laboratory (ARL) under Cooperative Agreement number DAAD19-01-2-0014. The content of which does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred. Access to research and computing facilities was provided by the Digital Technology Center and the Minnesota Supercomputing Institute.

ate feature space, represent each sequence as a vector in that space, and then take the inner product (or a function derived from them) between these vector-space representations as a kernel for the sequences. One of the early attempts with such feature-space-based approaches is the SVM-Fisher method [12], in which a profile HMM model is estimated on a set of proteins belonging to the positive class and used to extract a vector representation for each protein. Another approach is the SVM-pairwise scheme [19], which represents each sequence as a vector of pairwise similarity between all sequences in the training set. The similarity between each pair of sequences (i.e., the value along each dimension) is computed as the E-value of the Smith-Waterman alignment score between them. A relatively simple feature space, containing all possible short subsequences ranging from 3–8 amino acids ($k$mers) is explored in a series of papers (Spectrum kernel [17], Mismatch kernel [18], and Profile kernel [16]). Despite the simplicity of the feature space, the resulting methods have been shown to produce very good results. The difference between these schemes is on the method used to represent each sequence. The Spectrum kernel represents each sequence as a 0/1 vector based on whether or not it contains the $k$mer corresponding to each dimension of the feature space. The Mismatch kernel allows for some degree of tolerance when determining if a particular dimension is present in the sequence or not. For each $k$mer u in the protein sequence, it sets to one all the dimensions of the feature space that correspond to $k$mers that differ in at most a predefined number of positions. Finally, the Profile kernel extends the ideas of the Mismatch kernel by generating a position specific scoring matrix for each protein sequence and utilizing it to determine whether or not a particular dimension is sufficiently similar to a protein sequence $k$mer. Specifically, for each $k$mer u in the protein sequence, a set of other $k$mers is generated whose profile-based ungapped alignment score with u is above a user-supplied threshold, and u is "subscribed" to all of the dimensions in that set. An entirely different feature space is explored by the SVM-Isites [10] and SVM-HMMSTR [11] methods that take advantage of a set of local structural motifs (SVM-Isites) and their relationships (SVM-HMMSTR).

An alternative to measuring pairwise similarity through a dot-product of vector representations is to calculate an explicit protein similarity measure. The recently developed LA-Kernel method [26] represents one such example of a *direct kernel function*. This scheme measures the similarity between a pair of protein sequences by taking into account all the optimal local alignment scores with gaps between all of their possible subsequences. The experiments presented in [26] show that this kernel is superior to previously developed schemes that do not take into account sequence profiles and that the overall classification performance improves by taking into account all local alignments.

In this paper we develop new kernel functions that are derived directly from explicit similarity measures and utilize sequence profiles. We present two classes of such kernel functions. The first class, referred to as window-based, determines the similarity between a pair of sequences by using different schemes to combine ungapped alignment scores of certain fixed-length subsequences. The second, referred to as local alignment-based, determines the similarity between a pair of sequences using Smith-Waterman alignments and a position independent affine gap model, optimized for the characteristics of the scoring system. Both kernel-classes utilize profiles constructed automatically via PSI-BLAST and employ a profile-to-profile scoring scheme we develop by extending a recently introduced profile alignment method [21].

Experiments on two benchmarks derived from SCOP, one designed to detect remote homologs and the other designed to identify folds, show that these new kernels produce results that are substantially better than those produced by all other state-of-the-art SVM-based methods. In addition, the experiments show that these newly proposed kernels, even when used in the absence of profiles, produce results that are better than those produced by existing non-profile based schemes.

2 Methods and Algorithms

2.1 SVM and Kernel Functions

Key to our algorithm for protein classification is its learning methodology, which is based on support vector machines. Given a set of positive training sequences $\mathcal{S}^+$ and a set of negative training sequences $\mathcal{S}^-$, an SVM learns a classification function $f(X)$ of the form

$$f(X) = \sum_{X_i \in \mathcal{S}^+} \lambda_i^+ \mathcal{K}(X, X_i) - \sum_{X_i \in \mathcal{S}^-} \lambda_i^- \mathcal{K}(X, X_i), \quad (1)$$

where λ_i^+ and λ_i^- are non-negative weights that are computed during training by maximizing a quadratic objective function, and $\mathcal{K}(.,.)$ is called the *kernel* function that is computed over the various training-set and test-set instances. Given this function, a new sequence X is predicted to be positive or negative depending on whether $f(X)$ is positive or negative. In addition, the value of $f(X)$ can be used to obtain a meaningful ranking of a set of instances, as it represents the strength by which they are members of the positive or negative class.

2.2 Sequence Profiles

The inputs to our classification algorithm are the various proteins and their profiles. A protein sequence X of length n is represented by a sequence of characters $X = \langle a_1, a_2, \ldots, a_n \rangle$ such that each character corresponds to one of the 20 standard amino acids. The profile of a protein X is derived by computing a multiple sequence alignment of X with a set of sequences $\{Y_1, \ldots, Y_m\}$ that have a statistically significant sequence similarity with X (i.e., they are sequence homologs). Many schemes have been developed for identifying the set of homologous sequences and computing

the multiple sequence alignment. In this paper we obtain the profiles using PSI-BLAST [1] as it combines both steps, is very fast, and has been shown to produce reasonably good results. However, the profile-based kernels developed here can be used with other methods of constructing sequence profiles as well.

The profile of a sequence X of length n is represented by two $n \times 20$ matrices. The first is its position-specific scoring matrix PSSM_X that is computed directly by PSI-BLAST using the scheme described in [1]. The rows of this matrix correspond to the various positions in X and the columns correspond to the 20 distinct amino acids. The second matrix is its position-specific *frequency* matrix PSFM_X that contains the frequencies used by PSI-BLAST to derive PSSM_X. These frequencies (also referred to as *target frequencies* [21]) contain both the sequence-weighted observed frequencies (also referred to as *effective frequencies* [21]) as well as the BLOSUM62 [9] derived-pseudocounts [1]. For each row, the frequencies were scaled so that they add up to one. In the cases in which PSI-BLAST could not produce meaningful alignments for certain positions of X, the corresponding rows of the two matrices were derived from the scores and frequencies of BLOSUM62.

2.3 Profile-based Sequence Similarity

Many different schemes have been developed for determining the similarity between profiles that combine information from the original sequence, position-specific scoring matrix, or position-specific target and/or effective frequencies [21, 30, 20]. In our work we use a scheme that is derived from PICASSO [8, 21]. Specifically, the similarity score between the ith position of protein's X profile, and the jth position of protein's Y profile is given by

$$S_{X,Y}(i,j) = \sum_{k=1}^{20} \text{PSFM}_X(i,k)\,\text{PSSM}_Y(j,k) + \sum_{k=1}^{20} \text{PSFM}_Y(j,k)\,\text{PSSM}_X(i,k), \quad (2)$$

where $\text{PSFM}_X(i,k)$ and $\text{PSSM}_X(i,k)$ ($\text{PSFM}_Y(j,k)$ and $\text{PSSM}_Y(j,k)$) are the values corresponding to the kth amino acid at the ith (jth) position of X's (Y's) position-specific score and frequency matrices.

Equation 2 determines the similarity between two profile positions by weighting the position-specific scores of one sequence according to the frequency at which the corresponding amino acid occurs in the second sequence's profile. Note that by construction, Equation 2 leads to a symmetric similarity score. The key difference between Equation 2 and the corresponding scheme used in [21] (referred to as PICASSO3), is that our measure uses the target frequencies, whereas the scheme of [21] was based on effective frequencies. Our experiments (not included here) indicate that target frequencies lead to better results.

2.4 Window-based Kernels

The first class of profile-based kernel functions that we developed determines the similarity between a pair of sequences by combining the ungapped alignment scores of certain fixed length subsequences (referred to as $w$mers). Given a sequence X of length n and a user-supplied parameter w, the $w\text{mer}$ at position i of X ($w < i \leq n - w$) is defined to be the $(2w+1)$-length subsequence of X centered at position i. That is, the $w\text{mer}$ contains x_i, the w amino acids before, and the w amino acids after x_i. We will denote this subsequence as $w\text{mer}_X(i)$.

Note that $w$mers are nothing more than the fixed-length windows used extensively in secondary structure prediction and in capturing local sequence information around a particular sequence position. Also, for some of the kernel functions described next, they also correspond to the $k$mers used by some of the feature-space derived kernel functions [17, 18, 16].

2.4.1 All Fixed-width $w$mers (AF-PSSM). The AF-PSSM kernel computes the similarity between a pair of sequences X and Y by adding-up the alignment scores of all possible $w$mers between X and Y that have a positive ungapped alignment score. Specifically, if the ungapped alignment score between two $w$mers at positions i and j of X and Y, respectively is denoted by $w\text{score}_{X,Y}(i,j)$, n and m are the lengths of X and Y, respectively, and $\mathcal{P}_w$ is the set of all possible $w\text{mer}$-pairs of X and Y with a positive ungapped alignment score, i.e,

$$\mathcal{P}_w = \{(w\text{mer}_X(i), w\text{mer}_Y(j)) \mid w\text{score}_{X,Y}(i,j) > 0\}, \quad (3)$$

for $w+1 \leq i \leq n-w$ and $w+1 \leq j \leq m-w$, then the AF-PSSM kernel computes the similarity between X and Y as

$$\text{AF-PSSM}_{X,Y}(w) = \sum_{(w\text{mer}_X(i), w\text{mer}_Y(j)) \in \mathcal{P}_w} w\text{score}_{X,Y}(i,j). \quad (4)$$

The ungapped alignment score between two $w$mers is computed using the profile-to-profile scoring method of Equation 2 as follows:

$$w\text{score}_{X,Y}(i,j) = \sum_{k=-w}^{w} S_{X,Y}(i+k, j+k). \quad (5)$$

Note that both the AF-PSSM kernel and the Profile kernel [16] determine the similarity between a pair of sequences by considering how all of their fixed-length subsequences are related in view of sequence profiles. However, unlike the feature-space based approach employed by Profile, the AF-PSSM kernels determine the $w\text{mer}$-based similarity of two sequences by comparing all of their possible $w$mers directly. This allows such kernels to precisely determine

whether two $w$mers are similar or not. In addition, compared to the neighborhood intersection-based scheme employed by Profile, by utilizing profile-based ungapped alignment scores the AF-PSSM kernel can provide better quantitative estimates of the degree to which two $w$mers are similar.

2.4.2 Best Fixed-width $w$mer (BF-PSSM).

In determining the similarity between a pair of sequences X and Y, the AF-PSSM kernel includes information about all possible $w\text{mer}$-level local alignments between them. In light of this observation, it can be thought of as a special case of the LA kernels proposed by Saigo *et al* [26], which compute the similarity between a pair of sequences as the sum of the optimal local alignment scores with gaps between all possible subsequences of X and Y.[1] The results reported in [26] show that taking into account all possible alignments leads to better results.

To see whether or not this is true in the context of the profile-derived $w\text{mer}$-based kernels, we developed a scheme that attempts to eliminate this multiplicity by computing the similarity between a pair of sequences based on a subset of the $w$mers used in the AF-PSSM kernel. Specifically, the BF-PSSM kernel selects a subset $\mathcal{P}'_w$ of $\mathcal{P}_w$ (as defined in Equation 3) such that (i) each position of X and each position of Y is present in at most one $w\text{mer}$-pair and (ii) the sum of the $w$scores of the selected pairs is maximized. Given $\mathcal{P}'_w$, the similarity between the pair of sequences is then computed as follows:

$$\text{BF-PSSM}_{X,Y}(w) = \sum_{(w\text{mer}(X,i),w\text{mer}(Y,j))\in\mathcal{P}'_w} w\text{score}_{X,Y}(i,j). \quad (6)$$

The way that BF-PSSM selects the $w$mers to be included in $\mathcal{P}'_w$ can be better understood if the possible $w\text{mer}$-pairs in $\mathcal{P}_w$ are viewed as forming an $n \times m$ matrix, whose rows correspond to the positions of X, columns to the positions of Y, and values correspond to the respective $w$scores. Within this context, $\mathcal{P}'_w$ corresponds to a matching of the rows and columns [24] whose weight is high (bipartite graph matching problem). Since the selection forms a matching, each position of X (or Y) contributes a single $w\text{mer}$ in Equation 6, and as such, eliminates the multiplicity present in the AF-PSSM kernel. At the same time, since we are interested in a highly weighted matching, we try to select the *best* $w$mers for each position. In our algorithm, we use a greedy algorithm to incrementally construct $\mathcal{P}'_w$ by including the highest weight $w$mers that are not in conflict with the $w$mers already in $\mathcal{P}'_w$.

Note that an alternate way of defining $\mathcal{P}'_w$ is to actually look for the maximum weight matching (i.e., the matching whose weight is the highest among all possible matchings). However, the complexity of the underlying bipartite maximum weight matching problem is relatively high ($O(n^2m + nm^2)$ [24]), and for this reason we use the greedy approach.

2.4.3 Best Variable-width $w$mer (BV-PSSM).

In fixed-width $w\text{mer}$-based kernels the width of the $w$mers is fixed for all pairs of sequences and throughout the entire sequence. As a result, if w is set to a relatively high value, it may fail to identify positive scoring subsequences whose length is smaller than $2w + 1$, whereas if it is set too low, it may fail to reward sequence-pairs that have relative long similar subsequences.

To overcome this problem, we developed a kernel, referred to as BV-PSSM, which is derived from the BF-PSSM kernel but operates with variable width $w$mers. In particular, given a user-supplied width w, it considers the set of all possible $w\text{mer}$-pairs whose length ranges from one to w, i.e.,

$$\mathcal{P}_{1\ldots w} = \mathcal{P}_1 \cup \ldots \cup \mathcal{P}_w, \quad (7)$$

and among them, it uses the greedy scheme employed by BF-PSSM to select a subset $\mathcal{P}'_{1\ldots w}$ of $w\text{mer}$-pairs that form a high weight matching. The similarity between the pair of sequences is then computed as follows:

$$\text{BV-PSSM}_{X,Y}(w) = \sum_{(w\text{mer}(X,i),w\text{mer}(Y,j))\in\mathcal{P}'_{1\ldots w}} w\text{score}_{X,Y}(i,j). \quad (8)$$

Since for each position of X (and Y), $\mathcal{P}'_{1\ldots w}$ is constructed by including the highest scoring $w\text{mer}$ for i that does not conflict with the previous selections, this scheme can automatically select the highest scoring $w\text{mer}$ whose length can vary from one up to w; thus, achieving the desired effect.

2.5 Local Alignment-based Kernels (SW-PSSM)

The second class of profile-based kernels that we examine compute the similarity between a pair of sequences X and Y by finding an optimal alignment between them that optimizes a particular scoring function. There are three general classes of optimal alignment-based schemes that are commonly used to compare protein sequences. These are based on global, local, and global-local (also known as end-space free) alignments [7]. Our experiments with all of these schemes indicate that those based on optimal local alignments (also referred to as Smith-Waterman alignments [28]) tend to produce somewhat better results. For this reason we use this method to derive a profile-based alignment kernel, which is referred to as SW-PSSM.

Given two sequences X and Y of lengths n and m, respectively, the SW-PSSM kernel computes their similarity as the score of the optimal local alignment in which the similarity between two sequence positions is determined using the profile-to-profile scoring scheme of Equation 2, and a position independent affine gap model. The actual alignment is

[1] The major differences are that the AF-PSSM kernel is profile-aware, only considers fixed-length $w$mers, and uses ungapped alignments.

computed using the $O(nm)$ dynamic programming algorithm developed by Gotoh [4].

Within this local alignment framework, the similarity score between a pair of sequences depends on the particular values of the affine gap model (i.e., gap-opening (*go*) and gap-extension (*ge*) costs) and the intrinsic characteristics of the profile-to-profile scoring scheme. In order to obtain meaningful local alignments, the scoring scheme that is used should produce alignments whose score must on average be negative with the maximum score being positive [28]. A scoring system whose average score is positive will tend to produce very long alignments, potentially covering segments of low biologically relevant similarity. On the other hand, if the scoring system cannot easily produce alignments with positive scores, then it may fail to identify any non-empty similar subsequences.

To ensure that the SW-PSSM kernel can correctly account for the characteristics of the scoring system, we modify the profile-to-profile scores calculated from Equation 2 by adding a constant value. This scheme, commonly referred to as *zero-shifting* [30], ensures that the resulting alignments have scores that on the average are negative while allowing for positive maximum scores. In our scheme, the amount of zero-shifting, denoted by *zs*, is kept fixed for all pairs of sequences, as a limited number of experiments with sequence-pair specific *zs* values did not produce any better results.

2.6 From Similarity Measures to Mercer Kernels

Any function can be used as a kernel as long as for any number n and any possible set of distinct sequences $\{X_1, \ldots, X_n\}$, the $n \times n$ Gram matrix defined by $K_{i,j} = \mathcal{K}(X_i, X_j)$ is symmetric positive semidefinite. These functions are said to satisfy Mercer's conditions and are called Mercer kernels, or simply valid kernels.

The similarity based functions described in the previous sections can be used as kernel functions by setting $\mathcal{K}(X_i, X_j)$ to be equal to one of AF-PSSM$_{X_i,X_j}$, BF-PSSM$_{X_i,X_j}$, BV-PSSM$_{X_i,X_j}$, or SW-PSSM$_{X_i,X_j}$. However, the resulting functions will not necessarily lead to valid Mercer kernels.

To overcome this problem we used the approach described in [26] to convert a symmetric function defined on the training set instances into positive definite by adding to the diagonal of the training Gram matrix a sufficiently large non-negative constant. Specifically, for each similarity-based training Gram matrix, we found its smallest negative eigenvalue and subtracted it from the diagonal. The resulting kernel matrix is identical to the similarity-based Gram matrix at all positions expect those along the main diagonal. We also experimented with the empirical kernel map approach proposed in [27], but we find that the eigenvalue-based scheme produced superior results.

3 Experimental Design

3.1 Dataset Description

We evaluated the classification performance of the profile-based kernels on a set of protein sequences obtained from the SCOP (Structural Classification of Proteins) database [22]. We formulated two different classification problems. The first was designed to evaluate the performance of the algorithms for the problem of homology detection when the sequences have low sequence similarities (i.e., the remote homology detection problem), whereas the second was designed to evaluate the extent to which the profile-based kernels can be used to identify the correct fold when there are no apparent sequence similarities (i.e., the fold detection problem).

3.1.1 Remote Homology Detection (Superfamily Detection). Within the context of the SCOP database, remote homology detection was simulated by formulating it as a superfamily classification problem. The same dataset and classification problems [2] have been used in a number of earlier studies [19, 11, 26] allowing us to perform direct comparisons on the relative performance of the various schemes. The data consisted of 4352 sequences from SCOP version 1.53 extracted from the Astral database, grouped into families and superfamilies. The dataset was processed so that it does not contain any sequence pairs with an E-value threshold smaller than 10^{-25}. For each family, the protein domains within the family were considered positive test examples, and protein domains within the superfamily but outside the family were considered positive training examples. This yielded 54 families with at least 10 positive training examples and 5 positive test examples. Negative examples for the family were chosen from outside of the positive sequences' fold, and were randomly split into training and test sets in the same ratio as the positive examples.

3.1.2 Fold Detection. Employing the same dataset and overall methodology as in remote homology detection, we simulated fold detection by formulating as a fold classification within the context of SCOP's hierarchical classification scheme. In this setting, protein domains within the same superfamily were considered to be as positive test examples, and protein domains within the same fold but outside the superfamily were considered as positive training examples. This yielded 23 superfamilies with at least 10 positive training and 5 positive test examples. Negative examples for the superfamily were chosen from outside of the positive sequences' fold and split equally into test and training sets[3]. Since the positive test and training instances were members of different superfamilies within the same fold, this new prob-

[2] The dataset and classification problem definitions are available at http://www.cs.columbia.edu/compbio/svm-pairwise.

[3] The classification problem definitions are available at http://bioinfo.cs.umn.edu/supplements/remote-homology/.

lem is significantly harder than remote homology detection, as the sequences in the different superfamilies did not have any apparent sequence similarity [22].

3.2 Profile Generation

The position specific score and frequency matrices used by the profile-based scoring method of Equation 2 were generated using the latest version of the PSI-BLAST algorithm (available in NCBI's blast release 2.2.10), and were derived from the multiple sequence alignment constructed after five iterations using an e value of 10^{-3} (i.e., we used `blastpgp -j 5 -e 0.001`). The PSI-BLAST was performed against NCBI's nr database that was downloaded in November of 2004 and contained 2,171,938 sequences.

3.3 SVM Learning

We use the publicly available support vector machine tool SVM^{light} [13] that implements an efficient soft margin optimization algorithm. Following the approach used by the LA-Kernel [26], for any given positive semi-definite kernel Gram matrix $\mathcal{K}(.,.)$ to be tested, we first normalize the points to unit norm in the feature space and separate them from the origin by adding a constant, that is, we construct the kernel

$$\mathcal{K}'(X,Y) = \frac{\mathcal{K}(X,Y)}{\sqrt{\mathcal{K}(X,X)\,\mathcal{K}(Y,Y)}} + 1, \tag{9}$$

which is then provided as input to SVM^{light}. Note that unlike previous work [12, 19, 26], we do not perform any additional class-dependent kernel regularization to account for classes of different size. Thus, the results reported for the kernels that we developed can potentially be further improved after such regularizations.

3.4 Evaluation Methodology

We measured the quality of the methods by using the receiver operating characteristic (ROC) scores, the ROC50 scores, and the median rate of false positives (mRFP). The ROC score is the normalized area under a curve that plots true positives against false positives for different possible thresholds for classification [6]. The ROC50 score is the area under the ROC curve up to the first 50 false positives. Finally, the mRFP is the number of false positives scoring as high or better than the median-scoring true positives.

Among these evaluation metrics, due to the fact that the positive class is substantially smaller than the negative class, the ROC50 is considered to be the most useful measure of performance for real-world applications [6]. For this reason, our discussions in the rest of this section will primary focus on ROC50-based comparisons.

Table 1: Comparative performance of the window-based kernel functions that rely on sequence profiles.

	Superfamily-level			Fold-level		
Kernel	ROC	ROC50	mRFP	ROC	ROC50	mRFP
AF-PSSM (1)	0.965	0.692	0.022	0.851	0.275	0.143
AF-PSSM (2)	0.978	0.816	0.013	0.909	0.338	0.075
AF-PSSM (3)	0.976	**0.833**	0.014	0.904	0.340	0.080
AF-PSSM (4)	0.956	0.816	0.019	0.911	**0.374**	0.067
BF-PSSM (1)	0.967	0.794	0.025	0.906	0.359	0.082
BF-PSSM (2)	0.980	**0.854**	0.015	0.928	**0.419**	0.059
BF-PSSM (3)	0.977	0.853	0.016	0.918	0.408	0.069
BF-PSSM (4)	0.965	0.830	0.031	0.918	0.414	0.060
BV-PSSM (1)	0.965	0.808	0.027	0.900	0.423	0.088
BV-PSSM (2)	0.973	**0.855**	0.018	0.927	0.475	0.052
BV-PSSM (3)	0.966	0.851	0.022	0.936	0.480	0.046
BV-PSSM (4)	0.963	0.850	0.026	0.941	**0.481**	0.043

The parameter associated with each kernel corresponds to the width of the $w\mathrm{mer}$ used to define the kernel. The ROC50 of the best performing value of w for each kernel is shown in bold, and the overall best ROC50 is also underlined.

4 Results

4.1 Performance of the Window-based Kernels

Table 1 summarizes the performance achieved by the window-based kernels for the superfamily- and fold-level classification problems across a range of w values.

These results show that for both the superfamily- and fold-level classification problems, the BV-PSSM kernel achieves the best results, the AF-PSSM kernel tends to perform the worst, whereas the BF-PSSM kernel's performance is between these two. In the case of superfamily classification, the performance advantage of BV-PSSM over that of BF-PSSM is relatively small, whereas in the case of fold classification, the former has a clear advantage. It achieves an ROC50 value that is on average 16.3% better accross the different window lengths.

Comparing the sensitivity of the three schemes based on the value of w, we see that, as expected, their performance is worse for $w = 1$, as they only consider $w$mers of length 3, and their performance improves as the value of w increases. In general, the BV-PSSM kernel performs better for larger windows, whereas the performance of the other kernels tends to degrade more rapidly as the length of the window increases beyond a point. Again, this result is consistent with the design motivation behind the BF-PSSM kernel. Also, the results show that the best value of w is also dependent on the particular classification problem. For most kernels, the best results for fold classification were obtained with longer windows compared to the superfamily classification.

To see the effect of using sequence profiles, we performed a sequence of classification experiments in which we used the same set of window-based kernel functions, but instead of scoring the similarity between two amino acids using the profile-based scheme (Equation 2), we used the

Table 2: Comparative performance of the window-based kernel functions that rely on BLOSUM62.

	Superfamily-level			Fold-level		
Kernel	ROC	ROC50	mRFP	ROC	ROC50	mRFP
AF-GSM (1)	0.906	0.403	0.068	0.720	0.093	0.288
AF-GSM (2)	0.921	0.461	0.055	0.739	0.118	0.255
AF-GSM (6)	0.926	0.549	0.048	0.770	**0.197**	0.217
AF-GSM (7)	0.923	**0.557**	0.056	0.777	0.192	0.210
BF-GSM (1)	0.904	0.488	0.071	0.803	0.166	0.177
BF-GSM (2)	0.923	0.584	0.064	0.808	0.189	0.162
BF-GSM (6)	0.934	**0.669**	0.053	0.822	**0.240**	0.157
BF-GSM (7)	0.933	0.665	0.056	0.812	0.236	0.178
BV-GSM (1)	0.906	0.486	0.070	0.808	0.167	0.176
BV-GSM (2)	0.919	0.571	0.064	0.808	0.182	0.166
BV-GSM (6)	0.930	**0.666**	0.052	0.840	0.242	0.140
BV-GSM (7)	0.929	0.658	0.054	0.845	**0.244**	0.133

The parameter associated with each kernel corresponds to the width of the $w\mathrm{mer}$ used to define the kernel. The ROC50 of the best performing value of w for each kernel is shown in bold, and the overall best ROC50 is also underlined.

BLOSUM62 position-independent scoring matrix. The results obtained from these experiments are summarized in Table 2. In this table, AF-GSM, BF-GSM, and BV-GSM refer to the BLOSUM62-variants of the corresponding window-based kernels (GSM stands for *global scoring matrix*).

These results clearly illustrate the advantage of using sequence profiles in designing kernel functions for both remote homology detection and fold recognition. The profile-based kernel functions achieve significant improvements over their non-profile counterparts across all different kernel functions, classification problems, and metrics.

Comparing the performance of the profile-based kernel functions across the two classification problems, we see that their overall effectiveness in remote homology detection (superfamily-level classification) is much higher than that of fold recognition. This result is in line with the underlying complexity of the classification problem, as the sequence-based signals for fold recognition are extremely weak. This is also manifested by the relative improvement achieved by the profile-based kernel functions over their BLOSUM62-based counterparts (Tables 1 and 2). For fold recognition, the ROC50 values of the profile-based kernels are higher than those based on BLOSUM62 by a factor of two, whereas for remote homology prediction, the relative ROC50 values are higher by 25%–30%.

In light of the previously published results on LA-Kernels [26], the better results achieved by the BF-PSSM and BV-PSSM kernels over those achieved by the AF-PSSM kernel (which also hold for their corresponding BLOSUM62-based instances of these kernels) were surprising. One explanation for this discrepancy may be the fact that our window-based kernels consider only short-length ungapped alignments, and the results may be different when longer alignments with gaps are considered as well.

Table 3: Comparative performance of the local alignment-based kernel functions that rely on sequence profiles.

	Superfamily-level			Fold-level		
Kernel	ROC	ROC50	mRFP	ROC	ROC50	mRFP
2.0, 0.125, 0.0	0.972	0.784	0.014	0.867	0.377	0.111
2.0, 0.250, 0.0	0.972	0.791	0.014	0.873	0.334	0.114
3.0, 0.125, 0.0	0.971	0.796	0.013	0.860	0.382	0.133
3.0, 0.250, 0.0	0.960	0.771	0.027	0.852	0.395	0.138
3.0, 0.750, 1.5	0.982	**0.904**	0.015	0.933	0.530	0.052
3.0, 0.750, 2.0	0.979	0.901	0.017	0.936	**0.571**	0.054

The three parameters for each kernel correspond to the values for the gap opening, gap extension, and zero-shift parameters, respectively. The ROC50 of the best performing scheme is underlined.

4.2 Performance of the Local Alignment-based Kernels

Table 3 summarizes the performance achieved by the optimal local alignment-based kernel for the superfamily- and fold-level classification problems across a representative set of values for the gap-opening, gap-extension, and zero-shift parameters. These parameter values were selected after performing a study in which the impact of a large number of value combinations was experimentally studied, and represent some of the best performing combinations. Due to space constraints, this parameter study is not included in this paper.

The most striking observation from these results is the major impact that the zero-shift parameter has to the overall classification performance. For both the superfamily- and fold-level classification problems, the best results are obtained by the SW-PSSM kernel for which the zero shift parameter has been considered and optimized (i.e., the results corresponding to the last two rows of Table 3).

Comparing the classification performance of the SW-PSSM kernel against the window-based kernels (Table 1) we see that the zero-shift optimized SW-PSSM kernel leads to better results than those obtained by the window-based kernels. Moreover, the relative performance advantage of SW-PSSM is higher for fold recognition over the superfamily classification problem. However, if the SW-PSSM kernel does not optimize the zero-shift parameter (*i.e.*, $zs = 0.0$), the window-based kernels consistently outperform the SW-PSSM kernel. We also performed a limited number of experiments to see the extent to which the performance of the window-based kernels can be improved by explicitly optimizing the zero-shift parameter for them as well. Our preliminary results show that these kernels are not significantly affected by such optimizations. However, we are in the process of further investigating its impact.

To also see the impact of sequence profiles in the context of kernels derived from optimal local alignments, we evaluated the classification performance of a set of kernel functions that compute the optimal local sequence alignment using the BLOSUM45 and BLOSUM62 amino acid scoring matrices. Table 4 shows some of the results obtained with these kernel

Table 4: Comparative performance of the local alignment-based kernel functions that rely on BLOSUM45 and BLOSUM62.

	Superfamily-level			Fold-level		
Kernel	ROC	ROC50	mRFP	ROC	ROC50	mRFP
B45, 3.0, 0.0	0.944	0.686	0.037	0.809	0.165	0.169
B45, 10.0, 0.0	0.940	0.687	0.042	0.789	0.200	0.185
B62, 3.0, 0.0	0.947	0.686	0.038	0.781	0.188	0.217
B62, 10.0, 0.0	0.912	0.599	0.060	0.781	0.182	0.185
B62, 5.0, 0.5	0.948	**0.711**	0.039	0.826	**0.223**	0.176
B62, 5.0, 1.0	0.946	**0.711**	0.038	0.808	0.214	0.155

The three parameters for each kernel correspond to the particular global scoring matrix (B45 for BLOSUM45 and B62 for BLOSUM62) and the values for the gap opening and zero-shift parameters, respectively. In all cases, the gap extension cost was set to 1.0. The ROC50 of the best performing scheme is underlined.

functions for a representative set of values for the gap opening, gap extension, and zero-shift parameters.

Comparing the results of Table 4 with those of Table 3 we see that, as was the case with the window-based kernels, incorporating profile information leads to significant improvements in the overall classification performance. In addition, these results show that (i) the widely used value for the gap-opening cost ($go = 10$) is not necessarily the best for either remote homology detection or fold recognition, and (ii) the classification performance achieved by local alignment kernels derived from the BLOSUM matrices can be further improved by explicitly optimizing the zero-shift parameter as well.

4.3 Comparisons with Other Schemes

Tables 5 and 6 compare the performance of the various kernel functions developed in this paper against that achieved by a number of previously developed schemes for the superfamily- and fold-level classification problems, respectively. In the case of the superfamily-level classification problem, the performance is compared against SVM-Fisher [12], SVM-Pairwise [19], and different instances of the LA-Kernel [26], SVM-HMMSTR [11], Mismatch [18], and Profile [16]. In the case of the fold-level classification problem, we only include results for the LA-Kernel and Profile schemes, as these results could be easily obtained from the publicly available data and programs for these schemes. (Obtaining comparative performance numbers for the other kernel functions is currently under way.)

The results in these tables show that both the window- and local alignment-based kernels derived from sequence profiles (i.e., AF-PSSM, BF-PSSM, BV-PSSM, and SW-PSSM) lead to results that are in general better than those obtained by existing schemes. Comparing the ROC50 values obtained by our schemes, we see that each one of them outperforms all existing schemes. The performance advantage of these kernels is greater over existing schemes that rely on sequence information alone (e.g., SVM-Pairwise, LA-Kernels), but still remains significant when compared against

Table 5: Comparison against different schemes for the superfamily-level classification problem.

Kernel	ROC	ROC50	mRFP
SVM-Fisher	0.773	0.250	0.204
SVM-Pairwise	0.896	0.464	0.084
LA-eig($\beta = 0.2$)	0.923	0.661	0.064
LA-eig($\beta = 0.5$)	0.925	0.649	0.054
SVM-HMMSTR-Ave	–	0.640	0.038
Mismatch	0.872	0.400	0.084
Profile(4,6)	0.974	0.756	0.013
Profile(5,7.5)	0.980	0.794	0.010
AF-PSSM(2)	0.978	0.816	0.013
BF-PSSM(2)	0.980	0.854	0.015
BV-PSSM(2)	0.973	0.855	0.018
SW-PSSM(3.0,0.750,1.50)	0.982	**0.904**	0.015
AF-GSM(6)	0.926	0.549	0.048
BF-GSM(6)	0.934	0.669	0.053
BV-GSM(6)	0.930	0.666	0.052
SW-GSM(B62,5.0,1,0.5)	0.948	0.711	0.039

The SVM-Fisher, SVM-Pairwise, LA-Kernel, and Mismatch results were obtained from [26]. The SVM-HMMSTR results were obtained from [11] and correspond to the best-performing scheme (the authors did not report ROC values). The Profile results were obtained locally by running the publicly available implementation of the scheme obtained from the authors. The ROC50 value of the best performing scheme has been underlined.

schemes that either directly take into account profile information (e.g., SVM-Fisher, Profile) or utilize higher-level features derived by analyzing sequence-structure information (e.g., SVM-HMMSTR). Also, the relative advantage of our profile-based methods over existing schemes is greater for the much harder fold-level classification problem over the superfamily-level classification problem. For example, the SW-PSSM scheme achieves ROC50 values that are 13.8% and 81.8% better than the best values achieved by existing schemes for the superfamily- and fold-level classification problems, respectively.

To get a better understanding of the relative performance of the various schemes across the different classes, Figures 1 and 2 plot the number of classes whose ROC50 was greater than a given threshold that ranges from 0 to 1. Specifically, Figure 1 shows the results for the remote homology detection problem, whereas Figure 2 shows the results for the fold detection problem. (Note that these figures contain only results for the schemes that we were able to run locally). These results show that our profile-based methods lead to higher ROC50 values for a greater number of classes than either the Profile or LA-kernels, especially for larger ROC50 values (e.g. in the range of 0.6 to 0.95). Also, the SW-PSSM tends to consistently outperform the rest of the profile-based direct kernel methods.

In addition, the results for the BF-GSM, BV-GSM, and SW-GSM kernels that rely on the BLOSUM scoring matrices show that these kernel functions are capable of producing results that are superior to all of the existing non-profile-based schemes. In particular, the properly optimized SW-GSM scheme is able to achieve significant improvements over the

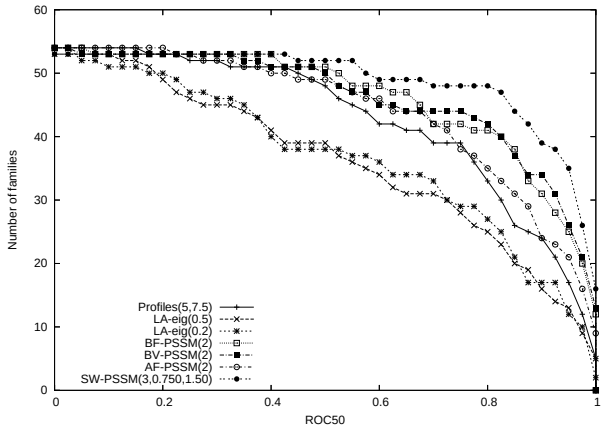

Figure 1: Comparison of the different SVM-based methods for remote homology detection on the SCOP 1.53 benchmark dataset. The graph plots the total number of families for which a given method exceeds an ROC-50 score threshold.

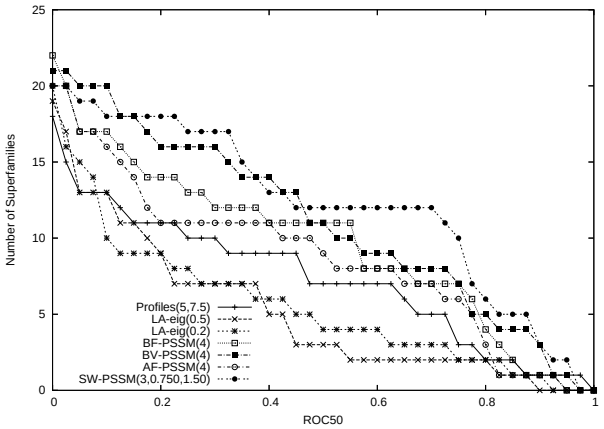

Figure 2: Comparison of the different SVM-based methods for fold detection on the SCOP 1.53 benchmark dataset. The graph plots the total number of superfamilies for which a given method exceeds an ROC-50 score threshold.

Table 6: Comparison against different schemes for the fold-level classification problem.

Kernel	ROC	ROC50	mRFP
LA-eig($\beta = 0.2$)	0.847	0.212	0.129
LA-eig($\beta = 0.5$)	0.771	0.172	0.193
Profile(4,6)	0.912	0.305	0.071
Profile(5,7.5)	0.924	0.314	0.069
AF-PSSM(4)	0.911	0.374	0.067
BF-PSSM(4)	0.918	0.414	0.060
BV-PSSM(4)	0.941	0.481	0.043
SW-PSSM(3.0,0.750,2.0)	0.936	**0.571**	0.054
AF-GSM(6)	0.770	0.197	0.217
BF-GSM(6)	0.822	0.240	0.157
BV-GSM(7)	0.845	0.244	0.133
SW-GSM(B62,5,1.0,0.5)	0.826	0.223	0.176

The results for the LA-Kernel were obtained using the publicly available kernel matrices that are available at the author's website. The Profile results were obtained locally by running the publicly available implementation of the scheme obtained from the authors. The ROC50 value of the best performing scheme has been underlined.

best LA-Kernel-based scheme (7.6% higher ROC50 value) and the best SVM-HMMSTR-based scheme (15.1% higher ROC50 value).

5 Discussion and Conclusion

This paper presented and experimentally evaluated a number of kernel functions for protein sequence classification that were derived by considering explicit measures of profile-to-profile sequence similarity. The experimental evaluation in the context of a remote homology prediction problem and a fold recognition problem show that these kernels are capable of producing superior classification performance over that produced by earlier schemes.

Three major observations can be made by analyzing the performance achieved by the various kernel functions presented in this paper. First, as was the case with a number of studies on the accuracy of protein sequence alignment [21, 30, 20], the proper use of sequence profiles lead to dramatic improvements in the overall ability to detect remote homologs and identify proteins that share the same structural fold. Second, kernel functions that are constructed by directly taking into account the similarity between the various protein sequences tend to outperform schemes that are based on a feature-space representation (where each dimension of the space is constructed as one of k-possibilities in a k-residue long subsequence or using structural motifs (Isites) in the case of SVM-HMMSTR). This is especially evident by comparing the relative advantage of the window-based kernels over the Profile kernel. Third, time-tested methods for comparing protein sequences based on optimal local alignments (as well as global and local-global alignments), when properly optimized for the classification problem at hand, lead to kernel functions that are in general superior to those based on either short subsequences (e.g., Spectrum, Mismatch, Profile, or window-based kernel functions) or local structural motifs (e.g., SVM-HMMSTR). The fact that these widely used methods produce good results in the context of SVM-based classification is reassuring as to the validity of these approaches and their ability to capture biologically relevant information.

References

[1] S. F. Altschul, L. T. Madden, A. A. Schffer, 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(17):3389–402, 1997.

[2] Stephen Altschul, Warren Gish, Webb Miller, Eugene Myers, and David Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[3] P. Baldi, Y. Chauvin, T. Hunkapiller, and M. McClure. Hidden markov models of biological primary sequence information. *PNAS*, 91:1053–1063, 1994.

[4] O. Gotoh. An improved algorithm for matching biological sequences. *Journal of Molecular Biology*, 162:705–708, 1982.

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

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

[7] Dan Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, New York, NY, 1997.

[8] A. Heger and L. Holm. Picasso:generating a covering set of protein family profiles. *Bioinformatics*, 17(3):272–279, 2001.

[9] S. Henikoff and J. G. Henikoff. Amino acid subsitution matrices from protein blocks. *PNAS*, 89:10915–10919, 1992.

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

[11] Y. Hou, W. Hsu, M. L. Lee, and C. Bystroff. Remote homolog detection using local sequence-structure correlations. *Proteins:Structure,Function and Bioinformatics*, 57:518–530, 2004.

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

[13] T. Joachims. Making large-scale svm learning practical. *Advances in Kernel Methods - Support Vector Learning, B. Schlkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999*, 1999.

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

[15] 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.

[16] R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, and C. Leslie. Profile-based string kernels for remote homology detection and motif extraction. *Computational Systems Bioinformatics*, pages 152–160, 2004.

[17] C. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: A string kernel for svm protein classification. *Proceedings of the Pacific Symposium on Biocomputing*, pages 564–575, 2002.

[18] C. Leslie, E. Eskin, W. S. Noble, and J. Weston. Mismatch string kernels for svm protein classification. *Advances in Neural Information Processing Systems*, 20(4):467–476, 2003.

[19] L. Liao and W. S. Noble. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. *Proc. of the Intl. Conf. on Research in Computational Molecular Biology*, pages 225–232, 2002.

[20] M. Marti-Renom, M. Madhusudhan, and A. Sali. Alignment of protein sequences by their profiles. *Protein Science*, 13:1071–1087, 2004.

[21] D. Mittelman, R. Sadreyev, and N. Grishin. Probabilistic scoring measures for profile-profile comparison yield more accurate short seed alignments. *Bioinformatics*, 19(12):1531–1539, 2003.

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

[23] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453, 1970.

[24] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Englewood Cliffs, NJ, 1982.

[25] W. Pearson. Rapid and sensitive sequence comparisons with fastp and fasta. *Methods in Enzymology*, 183:63–98, 1990.

[26] H. Saigo, J. P. Vert, N. Ueda, and T. Akutsu. Protein homology detection using string alignment kernels. *Bioinformatics*, 20(11):1682–1689, 2004.

[27] Bernhard Scholkopf and Alexander Smola. *Learning with Kernels*. MIT Press, Boston, MA, 2002.

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

[29] V. Vapnik. *Statistical Learning Theory*. John Wiley, New York, 1998.

[30] G. Wang and R. L. Dunbrack JR. Scoring profile-to-profile sequence alignments. *Protein Science*, 13:1612–1626, 2004.