Structure-based Kernel for Remote Homology Detection

Ivaylo Ilinkin
Department of Computer Science
Gettysburg College
Gettysburg, PA 17325, USA
iilinkin@gettysburg.edu

Jieping Ye
Department of Computer Science and Engineering
Center of Evolutionary Functional Genomics
The Biodesign Institute
Arizona State University
Tempe, AZ 85287, USA
jieping.ye@asu.edu

Abstract
Remote homology detection is a central problem in computational biology. Currently, the most effective tools for addressing this problem are kernel-based discriminative methods employing support vector machines. These methods work by transforming the protein sequences into (a possibly high-dimensional) vector space, called feature space, and deriving a kernel function in the feature space, which is used to train a support vector machine. The effectiveness of these methods depends on the protein representation and the choice of feature space. In general, the current methods use sequence information for constructing a suitable feature space; the three-dimensional structure of proteins has been largely under-utilized in addressing the problem of remote homology detection.

This paper introduces a structure-based approach for remote homology detection. The approach considers contiguous fragments along the protein backbone and derives a feature representation for the protein based on the three-dimensional structure of the fragments. Protein similarity is measured in terms of the number of shared fragments with similar three-dimensional structure. In a pre-processing step the set of all fragments is organized into a fixed number of clusters; the feature vector computed for each protein represents the number of fragments that belong to each individual cluster. Experimental results demonstrate the viability of the approach and show that it is highly competitive in terms of accuracy with current methods for remote homology detection.
1 Introduction

In recent years there has been a considerable amount of research in developing new methods for classification of proteins in structural and functional families. These methods provide important tools for understanding the function of newly discovered proteins, since often the function of a new protein can be inferred by identifying related proteins whose properties are well understood.

When the proteins share high degree of sequence similarity, methods based on algorithms for pairwise sequence alignment \([1, 2, 3]\) perform well in detecting protein homologies. In the case of distantly related proteins, however, other approaches have proven to be more successful.

Generative methods based on hidden Markov models (HMM) \([4, 5, 6]\) build a model for a family of related proteins and check the degree to which a new protein fits the model. In general, these methods use only information from proteins within the family (positive examples), and do not incorporate in the model discriminative characteristics from proteins outside the family (negative examples).

Discriminative methods based on support vector machines (SVM) \([7]\) use information from both positive and negative examples to build a classifier, which can then be used to determine whether a new protein belongs to the protein family. These methods currently represent the state-of-the-art in protein classification. The accuracy of SVM classifiers depends on a suitable representation of the proteins in a (possibly high-dimensional, even infinite) feature space and a kernel function (or kernel for short) defined in the feature space. A sampling of recent work includes a kernel derived using a generative model based on HMMs \([8]\), a kernel derived from pairwise sequence similarity scores \([9]\), and a series of papers \([10, 11, 12, 13]\) that introduce string-based kernels which offer a computationally efficient alternative of comparable accuracy. Discriminative approaches that derive position-dependent kernels currently provide the best accuracy. Saigo et al \([14]\) propose local alignment kernels which measure the similarity between two sequences based on local alignments. Rangwala and Karypis \([15]\) develop direct kernels using sequence profiles that are constructed automatically via PSI-BLAST and employ a profile-to-profile scoring scheme. Most recently, Lingner and Meinicke \([16]\) introduce a feature representation of protein sequences based on distances between short oligomers and show that their method is competitive in terms of accuracy with previous approaches, but is more efficient computationally.

The state-of-the-art approaches for remote homology detection derive the feature space and kernel function based on sequence information. Structure information has been used recently to derive a kernel based on pairwise structure alignment \([17]\). The use of structure alignment in computing the kernel can incur high computational cost. In this paper we propose a structure-based kernel derived from the distance information in the three-dimensional structure of the protein backbone which gives rise to a feature vector representation of the protein in a high-dimensional space. Our experimental results demonstrate the viability of the approach in terms of accuracy and computational efficiency.
The rest of the paper is organized as follows. We provide basic background in Section 2. Our proposed structure-based kernel is presented in Section 3. Experimental results are given in Section 4. Discussion and directions for future research are offered in Section 5.

\section{Basics}

In this section, we give a brief overview of several relevant methods including Support Vector Machines (SVM) and string kernels.

\subsection{Support Vector Machines}

Support vector machines have been applied successfully for classification tasks in a number of domains. Given a vector of training examples

\[ X = (\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n), \vec{x}_i \in \mathbb{R}^n \]

and a vector of their corresponding labels

\[ Y = (y_1, y_2, \ldots, y_n), y_i \in \{\pm 1\} \]

an SVM solves an optimization problem to find a linear function

\[ f(\vec{x}) = \langle \vec{w}, \vec{x} \rangle + b \]

that correctly classifies the given examples using the sign of \( f \) as the decision rule: input \( \vec{x} \) is assigned to the positive class if \( f(\vec{x}) \geq 0 \); otherwise \( \vec{x} \) is assigned to the negative class.

In general, the representation of the data affects the complexity of the classification task. While the given examples may not be linearly separable in the original input space, linear separation may be possible after a transformation via a suitable non-linear map, \( \Phi \), into another vector space, called feature space.

In the dual version of the optimization problem the input data occurs only in the form of dot products \( \langle \vec{w}, \vec{x} \rangle \), and therefore, given a kernel function

\[ K(\vec{x}, \vec{y}) = \langle \Phi(\vec{w}), \Phi(\vec{x}) \rangle \]

the SVM can be trained to classify the examples in the new feature space. If the kernel function can be computed efficiently without explicit transformation of the data points in the feature space, the learning can take place into a high-dimensional (possibly infinite) feature space; moreover, an implicit representation of the data may result in significant computational efficiency.

Detailed treatment of support vector machines and kernel-based learning methods can be found in Christianini and Shawe-Taylor [18, 19].
2.1.1 String-kernels

The structure-based kernel is closely related to the string-kernels previously proposed for protein sequences [10, 11, 12, 13]. A kernel function can be thought of as a measure of similarity between sequences. Different kernels correspond to different notions of similarity measure, and can lead to discriminative functions with different performance. Polynomial kernels and Gaussian kernels are commonly used for vectorized data. For abstract data such as protein sequences a kernel is typically designed by choosing an appropriate vector representation for the sequences and then taking the inner product between these representations as a kernel.

Leslie et al [11] proposed a simple string kernel, called the spectrum kernel, which is very easy to compute. This kernel is based on the similarity between length-\(k\) substrings, called \(k\)-mers. For example, for \(k = 3\) and an alphabet of size 20, each protein is represented as a vector of length \(20^3 = 8000\), each corresponding to a single 3-mer. The kernel can be computed efficiently using a trie data structure. The spectrum kernel using \(k = 3\) has been shown to be comparable to that of the HMM-based Fisher kernel on the SCOP benchmark. The spectrum kernel has also been generalized to allow for a more accurate model of molecular evolution. Mutations in the protein sequence are modeled using a mismatch kernel [11] which allows a certain number of mismatches between the \(k\)-mers.

3 Proposed structure-based kernel

Our structure-based approach derives a feature space based on the distance information in the three-dimensional structure of the protein backbone. Following the key idea from string kernels, we consider all subsequences of fixed length \(m\) of consecutive \(C_\alpha\) atoms along the protein backbone. For each subsequence we compute the Euclidean distance between all pairs of its \(C_\alpha\) atoms and encode the distance information into a vector called an \(m\)-gram. Unlike the mismatch kernel, where there is only a finite number of distinct \(k\)-mers, the \(m\)-grams for two subsequences with similar structure are not going to be identical, in general. To obtain a feature representation for each protein structure, we partition the \(m\)-grams of all protein structures into \(k\) clusters that give rise to a \(k\)-dimensional feature space — the feature vector for each protein records in its \(i\)-th component the number of \(m\)-grams that belong to the \(i\)-th cluster.

We now describe each of the steps of the feature-space transformation in more detail:

3.1 \(m\)-gram computation

For each subsequence of \(m\) consecutive \(C_\alpha\) atoms, \(\{C_\alpha^{(i)}\}_{i=1}^m\), along the protein backbone we compute the corresponding distance matrix, which records the pairwise Euclidean distance between every pair of \(C_\alpha\) atoms. Let \((x_i, y_i, z_i)\) be
the 3D coordinate of the \(i\)-th \(C_{\alpha}\) atom. The distance matrix \(D = (d_{ij})\) is given by
\[
d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}.
\]
Since the distance matrix is symmetric and its diagonal elements are 0, we only keep the portion above the diagonal by listing the relevant elements column-wise in an \(m\)-gram vector of dimension \(\frac{m(m-1)}{2}\). The \(m\)-gram representation of a given protein is the set of all of its \(m\)-grams. The \(m\)-gram representation is orientation independent, since the \(m\)-grams encode only distance information.

### 3.2 Cluster computation

We partition the set of all \(m\)-grams of all of the training examples into \(k\) clusters using CLUTO.\(^1\) The output of CLUTO consists of the label of the cluster corresponding to each data point in the input. Based on this information we compute the centroids, \(\{c_l\}_{l=1}^k\), of all clusters and use the centroids as cluster representatives.

### 3.3 Feature computation

For each test protein we first compute its \(m\)-gram representation, \(\{\mu_i\}_{i=1}^n\), where \(n\) is the total number of \(m\)-grams. We then use the cluster centroids to derive a feature vector \(\phi\) of dimension \(k\). Each \(m\)-gram, \(\mu_i\), is assigned to the \(L_i\)-th cluster based on the Euclidean distance to its centroid as follows:
\[
l_i = \arg\min_l \|\mu_i - c_l\|.
\]
The feature vector \(\phi\) is then constructed by recording in each component \(\phi_i\) the number of \(m\)-grams assigned to the \(i\)-th cluster, that is
\[
\phi_j = \{|i : L_i = j\}|.
\]
The structure-based kernel is based on the inner product of the feature vector \(\phi\) constructed for all proteins.

### 4 Experiments and results

#### 4.1 Dataset description

Our experiments are based on a standard benchmark dataset [9] for evaluating the accuracy of methods for remote homology detection. In that benchmark the data is obtained from the Astral compendium based on version 1.53 of the SCOP database [20, 21] and is pre-processed so that no pair of sequences in the benchmark dataset has an \(E\)-value threshold \(< 10^{-25}\). The sequences within

\(^1\)http://glaros.dtc.umn.edu/gkhome/views/cluto
each family are considered positive test examples, while those outside the family but within the same superfamily are considered positive training examples. The negative examples are taken from sequences outside of the superfamily and are split randomly into test and training examples in the same ratio as the positive examples. The dataset consists of 4352 distinct sequences organized into 54 families with at least 10 positive training examples and 5 positive test examples.

We follow a similar approach in designing our experimental framework. One of the major differences is the fact that our dataset is based on version 1.69 of the SCOP database, which is the latest release at the time of this writing. The SCOP database has undergone significant revisions since version 1.53 in terms of the amount of available information, organization, and categorization of structures. Fortunately, it is still possible to make meaningful comparisons with other approaches that make available the software for computing the kernel matrix for a given dataset.

For our experiments we use the sequence information and PDB-style files for a dataset of sequences with less than 40 on version 1.69 of the SCOP database and is available through the Astral compendium at http://astral.berkeley.edu/pdbstyle-seq-gs-bib-40-1.69.tgz (PDB files) and at http://astral.berkeley.edu/ (sequence data). We use the method of Liao and Noble [9] to split the data into training and test examples. The final dataset consists of 7286 distinct sequences organized into 123 families with at least 10 positive training examples and 5 positive test examples.

4.2 Evaluation methodology

We have compared the accuracy of our structure-based kernel (SVM-structure) with the mismatch kernel [11] (SVM-mismatch). Since SVM-mismatch has been shown to be of comparable accuracy to the method of Liao and Noble [9] (SVM-pairwise), this also provides indirect comparison between SVM-structure and SVM-pairwise.

For our experiments we report the receiver operating characteristic (ROC) scores and the ROC50 scores. The ROC score is the normalized area under the curve that plots true positives against false positives for a varying decision threshold [22]. The ROC50 score is the area under the ROC curve plotted up to the first 50 false positives. Of these two metrics the ROC50 score is considered the more useful [23].

4.3 Experimental results

The experiments were run within the software framework for the string-based kernels [11, 12, 13] and Weston [10]. The software is available for public download at http://cbio.mskcc.org/leslielab/software/string-kernels/. It uses SPIDER for the SVM training and testing, which is available at http://www.kyb.tuebingen.mpg.de/bs/people/spider/.

We ran the code for the mismatch kernel with parameters (5, 1), which represent the fixed length of subsequences and the number of allowed mismatches,
respectively. This set of parameters is the one that achieves the best accuracy as reported in Leslie et al [11]. We also computed the feature vectors for our structure-based representation for \( m \)-grams of size \( m = 5, 7, 10 \), and number of clusters \( k = 1000; 2000; 3000; 10000 \). SVM-structure performed better than SVM-mismatch for all combinations of parameters; the highest accuracy was achieved for \((m, k) = (7, 3000)\).

Table 1 shows the results of our experiments. The table reports the two evaluation metrics — ROC and ROC50 — for each of the methods averaged over the 123 families in the experiments. Figure 1 summarizes graphically the results of the experiments by plotting the number of families that exceed a given ROC threshold. A similar summary is presented in Figure 2 which plots the number of families that exceed a given ROC50 threshold. The ROC50 is considered to be the more useful metric, in general [23]. Figure 3 shows a family-by-family comparison between the methods by plotting one point for each family whose \((x, y)\) coordinates are the ROC scores of SVM-structure and SVM-mismatch. The results show that SVM-structure outperforms significantly SVM-mismatch in terms of accuracy.

### Table 1: Summary of the classification results for each of the methods based on the evaluation metrics used for the experiments.

<table>
<thead>
<tr>
<th>Method</th>
<th>Average ROC</th>
<th>Average ROC50</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM-structure-(7, 3000)</td>
<td>0.9236</td>
<td>0.5207</td>
</tr>
<tr>
<td>SVM-mismatch-(5, 1)</td>
<td>0.8363</td>
<td>0.3131</td>
</tr>
</tbody>
</table>

A comparison of kernel-based methods needs to take into account the actual running time for computing the kernel matrix. The time for computing the kernel matrix by SVM-structure includes computing the \( m \)-grams for each of the proteins and the time for computing the \( k \) clusters. Since \( m \) is a fixed constant, the \( m \)-gram computation simply involves a linear scan of the protein backbone and is therefore very efficient. The overall running time is dominated by the time to compute the \( k \) clusters in the \( m \)-grams space. Excluding the time for training the SVM the total time taken by SVM-mismatch-(5, 1) for a single family was approximately 22 minutes; SVM-structure-(7, 3000) took approximately 39 minutes which includes 33 minutes for computing the 3000 clusters. The experiments were done on a Linux-based machine with 3.8GB of RAM and four 3.20GHz Intel(R) Xeon(TM) processors.

### 4.4 Combination of kernels

One of our experiments investigated the effectiveness of combining the sequence-based kernel of SVM-mismatch and the structure-based kernel of SVM-structure. We use a simple linear combination of kernels \( K(\vec{x}, \vec{y}) = K_1(\vec{x}, \vec{y}) + K_2(\vec{x}, \vec{y}) \), where \( K_1 \) and \( K_2 \) are the kernels for SVM-mismatch-(5, 1) and SVM-structure-(7, 3000), respectively. The results from this experiment are summarized in Figure 4 and Figure 5. SVM-mixed, which is based on a combination of kernels,
Figure 1: Comparison of SVM-structure-(7, 3000) and SVM-mismatch-(5, 1) based on the ROC score. Each curve plots the total number of families for which the corresponding method exceeds a given ROC threshold.

Figure 2: Comparison of SVM-structure-(7, 3000) and SVM-mismatch-(5, 1) based on the ROC50 score. Each curve plots the total number of families for which the corresponding method exceeds a given ROC50 threshold.

achieved the highest accuracy.

5 Discussion and conclusion

This paper has introduced a structure-based approach to the problem of remote homology detection. We build a discriminative model for superfamily classifi-
Figure 3: Family-by-family comparison between SVM-structure-(7, 3000) and SVM-mismatch-(5, 1). Each point represents the performance of both methods for one family. The \((x, y)\) coordinates of each point are the ROC scores for SVM-structure-(7, 3000) and SVM-mismatch-(5, 1), respectively.

Figure 4: Comparison of SVM-mixed with SVM-structure-(7, 3000) and SVM-mismatch-(5, 1) based on the ROC score. Each curve plots the total number of families for which the corresponding method exceeds a given ROC threshold.

cation based on support vector machines and a kernel matrix derived from the three-dimensional structure of the protein backbone. Our approach considers each sub-sequence of length \(m\) of consecutive \(C_{\alpha}\) atoms along the protein backbone and represents it as an \(m\)-gram vector that encodes the distance information between every pair of atoms. The transformation to feature space is
Figure 5: Comparison of SVM-mixed with SVM-structure-(7, 3000) and SVM-mismatch-(5, 1) based on the ROC50 score. Each curve plots the total number of families for which the corresponding method exceeds a given ROC50 threshold.

achieved by partitioning the \( m \)-grams of all training examples into \( k \) clusters and representing each protein as a \( k \)-dimensional vector whose \( i \)-th component records the number of \( m \)-grams that are assigned to the \( i \)-th cluster.

Our experimental results demonstrate that the structure-based method outperforms existing sequence-based methods in terms of two commonly used metrics — the receiver operating characteristic (ROC) score and the ROC50 score. Computationally the method is very efficient since it involves a linear scan through the backbone of each protein and a clustering step for which efficient implementations are available.

In future work we plan to investigate whether the trained SVM could yield insights about the structure of the dataset. For example, it would be useful to analyze the support vectors and identify if they carry meaningful interpretation (e.g. representatives for their respective families). Another important question is whether information from multiple structure alignment, specifically the structure of conserved regions, can be incorporated into our discriminative model. Our current approach considers protein similarities in terms of contiguous fragments along the backbone. We would like to incorporate in our model information about the relative relationship between structural fragments along the backbone. To this end it may be possible to consider \( m \)-grams derived from the complete distance matrix of the protein, but our preliminary investigation indicates that this comes at high computational cost.
References


