Learning Random-Walk Kernels for Protein Remote Homology Identification and Motif Discovery

Renqiang Min* Rui Kuang† Anthony Bonner‡ Zhaolei Zhang§

Abstract

Random-walk based algorithms are good choices for solving many classification problems with limited labeled data and a large amount of unlabeled data. However, it is difficult to choose the optimal number of random steps, and the results are very sensitive to the parameter chosen. In this paper, we will discuss how to better identify protein remote homology than any other algorithm using a learned random-walk kernel based on a positive linear combination of random-walk kernels with different random steps, which leads to a convex combination of kernels. The resulting kernel has much better prediction performance than the state-of-the-art profile kernel for protein remote homology identification. On the SCOP benchmark dataset, the overall mean ROC_{50} score on 54 protein families we obtained using the new kernel is above 0.90, which has almost perfect prediction performance on most of the 54 families and has significant improvement over the best published result; moreover, our approach based on learned random-walk kernels can effectively identify meaningful protein sequence motifs that are responsible for discriminating the memberships of protein sequences' remote homology in SCOP.

1 Introduction

Machine learning researchers are often faced with classification problems with limited labeled data and a large amount of unlabeled data. In biological problems, this is almost always the case. It takes long-time tedious human work or expensive biological experiments to label data. Like the protein remote homology problem we will describe here, we often have several positive training cases, many negative training cases, and a lot of unlabeled data for many protein families. Therefore, we need good algorithms that can best take advantage of the unlabeled data. Moreover, classifying biological sequences is an important and challenging problem both in computational biology and machine learning. On the biological side, it helps to identify interesting sequence regions and protein domains that are related to a particular biological function; on the computational side, it motivates many novel and effective new classification approaches specifically for sequence data. Generative models (e.g., profile HMMs [11], [3]), discriminative models (e.g., kernel SVMs [9], [14], [15]), and graph-based approach [22] have been applied to solve this problem.

In [9], [14], [23], [15] and [12], it has been shown that kernel SVMs have better prediction performance on biological sequence data than other methods. Moreover, it was shown in [23] that random-walk kernels ([19] and [6]) produced promising results on protein remote homology detection. However, the process of deciding the optimal number of random steps in a random-walk kernel remains as a challenging problem [23]. In this paper, we propose using label information of training data and a positive linear combination of random-walk kernels to approximate the random-walk kernel with the optimum steps of a random walk, thereby obtaining a convex combination of random-walk kernels with different random-walk steps which achieves the best classification confidence on the labeled training set.

As is described in [12], kernel SVMs can not only be applied to classify biological sequences, but also they can be used to extract discriminative sequence motifs that explain the classification results. In this paper, we will use SVMs based on learned random-walk kernels to extract protein sequence motifs contributing to discriminating protein sequences' remote homology. Experimental results on the SCOP benchmark dataset show that learned random-walk kernel achieves significant improvement over the best published result and the result given by the random-walk kernel with a fixed number of random steps, and it effectively extracts meaningful protein sequence motifs.

This paper is organized as follows: section 2 gives a brief introduction to SVM classification based on mismatch-string kernels. Section 3 describes our

*Department of Computer Science, University of Toronto, Canada.

†Department of Computer Science, University of Minnesota, Twin Cities, USA.

‡Department of Computer Science, University of Toronto, Canada.

§Banting and Best Department of Medical Research, University of Toronto, Canada

method of learning random-walk kernels, a positive linear combination of random-walk kernels. Section 4 describes protein sequence motif discovery using SVMs based on learned random-walk kernels. Section 5 present experimental results of protein homology detection and motif discovery on the SCOP dataset. Section 6 concludes the paper with a discussion on our proposed method and provides some ideas for future research.

2 SVM for biological sequence classification using mismatch string kernels

A SVM ([21] and [24]) is a discriminative model proposed especially for classification. Consider a two-class training set, $\{X, y\}$ and a test set U, where X is a matrix whose i-th column, X_i, is the feature vector of data point i in the training set, U is a matrix whose j-th column, U_j, is the feature vector of data point j in the test set, and y, a column vector whose i-th component y_i is the label of data point i in the labeled set, $y_i \in \{-1, 1\}$, $X_i, U_j \in R^d$, $i = 1, \cdots, N, j = 1, \cdots, M$. A linear SVM gives a separating hyper-plane that maximizes the margin between the sample data points of the two classes. The dual problem of a soft-margin SVM can be formulated as follows:

$$max_\alpha \quad 2\alpha^T \mathbf{1} - \alpha^T (K_{tr} \otimes yy^T)\alpha, s.t. \alpha^T y = 0, \mathbf{0} \leq \alpha \leq C\mathbf{1}, \tag{2.1}$$

where $\mathbf{1}$ and $\mathbf{0}$ are column vectors containing all ones and zeros respectively, $\otimes$ is the component-wise matrix multiplication operator, $K = [X|U]^T[X|U]$, is the dot product between feature vectors of pairwise data points, K_{tr} is the training part of K where $K_{tr} = X^T X$, and, C is the penalty coefficient penalizing margin violations. As the above dual problem is only dependent on dot-products between feature vectors, we can discard the original feature vectors of data points and calculate a kernel matrix K directly to represent the relationship between the original data points. As is discussed in [21], any symmetric positive semi-definite matrix can be used as a valid kernel matrix K. Therefore by constructing a kernel, K, we can map every data point, X_i, to a high-dimensional feature space, in which a SVM can be used to generate a separating hyper-plane.

For biological sequences, a kernel function can be used to map these sequences consisting of characters representing amino acids to a higher dimensional feature space on which a max-margin classifier is trained. All the computations of a SVM are performed on the dot products of the pairwise feature vectors stored in the kernel matrix. For example, suppose A is an alphabet of ℓ symbols ($\ell = 20$ for protein sequences), then k-mer string kernel maps every sequence in A to a ℓ^k-dimensional feature space in which coordinates are indexed by all possible sub-sequences of length k (k-mers). Specifically, the feature map of a k-mer string kernel is given by

$$\Phi_k(x) = \left(\Phi_{\alpha_1}(x), \Phi_{\alpha_2}(x), \cdots, \Phi_{\alpha_{l^k}}(x)\right)^T, \tag{2.2}$$

where $\alpha_1, \alpha_2, \ldots, \alpha_{\ell^k}$ is an ordering of all the ℓ^k possible k-mers, and $\Phi_\alpha(x)$ is the number of occurrences of k-mer α in sequence x. The corresponding kernel matrix is

$$K_k(x, y) = \Phi_k(x)^T \Phi_k(y). \tag{2.3}$$

The mismatch string kernel extends this idea by accommodating mismatches when counting the number of occurrences of a k-mer in an input sequence. In particular, for any k-mer, α, let $N_{(\alpha,m)}$ be the set of all k-mers that differ from α by at most m mismatches. The kernel mapping and kernel matrix are then defined as follows:

$$\Phi_{(k,m)}(x) = (\Phi_{(k,m),\alpha_1}(x), \cdots, \Phi_{(k,m),\alpha_{\ell^k}}(x))^T, \tag{2.4}$$

$$\Phi_{(k,m),\alpha}(x) = \sum_{\beta \in N_{(\alpha,m)}(x)} \Phi_\beta(x), \tag{2.5}$$

$$K_{(k,m)}(x, y) = \Phi_{(k,m)}(x)^T \Phi_{(k,m)}(y). \tag{2.6}$$

A profile of a protein sequence is a sequence of multinomial distributions. Each position of a protein sequence's profile is a multinomial distribution on 20 amino acids, representing the emission probabilities of the 20 amino acids at each position in that sequence. A Profile Kernel [12] extends the mismatch-string kernel by using additional profile information of each sequence. Instead of treating all k-mers with less than m mismatches similarly as the mismatch-string kernel described above, the profile-kernel examines these k-mers further by looking at the emission probabilities (profiles) at the mismatched positions and only accepts those mismatches that pass a certain threshold. The work-flow for constructing a profile kernel as described in [12] is shown in Fig. 1. Each sequence has a profile, which is obtained by iteratively aligning each sequence to the sequences in an unlabeled set using PSI-BLAST [1]. Suppose we have a sequence $x = x_1 x_2 ... x_N$ of amino acids of length N, then $P(x) = \{p_i^x(a), a \in \Sigma\}_{i=1}^N$ is the profile of sequence x, where Σ is the set of 20 amino acids and $p_i^x(\cdot)$ is the multinomial distribution on the 20 amino acids at the i-th position of the profile of sequence x. For e.g., $p_i^x(a)$, is the emission probability of amino acid a at position i, such that $\sum_{a \in \Sigma} p_i(a) = 1$ at each position i. In the Profile Kernel, the neighborhood of a

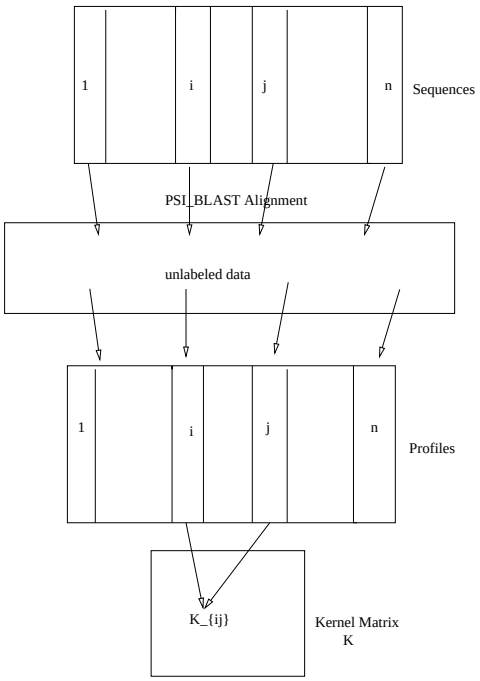

Figure 1: The work-flow of constructing a profile kernel in [12].

k-mer $x[j+1:j+k]$=$x_{j+1}x_{j+2}...x_{j+k}$ in sequence x is defined as:

$$M_{(k,\sigma)}(P(x[j+1:j+k])) = \{\beta = b_1...b_k : -\sum_{i=1}^{k} logp^x_{j+i}(b_i) < \sigma\}, \tag{2.7}$$

where the free parameter σ controls the size of the neighborhood, and $p^x_{j+i}(b)$ for $i = 1, ..., k$ is obtained from the profile of sequence x, $0 \leq j \leq |x| - k$. Further, $p^x_{j+i}(b)$ can be smoothed using the background frequency of amino acid b. The feature vector of sequence x in the Profile Kernel is defined as the following:

$$\Phi_{(k,\sigma)}(x) = \sum_{j=0}^{|x|-k} (\phi_{\beta_1}(P(x[j+1:j+k])), \ldots, \phi_{\beta_{\ell^k}}(P(x[j+1:j+k])))^T, \tag{2.8}$$

where $\beta_1, \ldots, \beta_{\ell^k}$ is an ordering of all possible k-mers, and, the coordinate $\phi_\beta(P(x[j+1:j+k]))$ is 1 if $\beta \in M_{(k,\sigma)}(P(x[j+1:j+k]))$, and 0 otherwise. The profile kernel uses the profile to measure the mismatch information between different letters at each position of each sequence. Therefore, it's more accurate than the mismatch string kernel. In this paper, we will use the profile kernel discussed above as the base kernel in the derivation of the random-walk kernel.

3 Random-walk kernel for biological sequence classification

In this section, we will describe learned random-walk kernels. As is discussed in section 1, we are often faced with classification problems with limited labeled data and a large amount of unlabeled data. These problems are often solved using similarity-propagation-based methods such as the method discussed in [25]. Random-walk based approaches are also examples of similarity-propagation based methods. Our motivation for using a random-walk kernel is its ability to coerce data points in the same cluster to stay closer while making data points in different clusters to stay farther apart by propagating similarity on both labeled data and unlabeled data (see [19] and [23]). If we view a set of data points as a complete (or sparse) graph, in which the weights between data points are viewed as similarity scores, then we can make use of unlabeled data to help propagate similarity information through the whole graph. For e.g., we have a graph containing two labeled data points, i and j, and two unlabeled data points, s and t, i is highly similar to s, s is highly similar to t, and t is highly similar to j, but i and j are not very similar to each other in the given graph. After two steps of similarity propagation, i and j will become similar in a new similarity graph. When the similarity-propagation process is over, we hope that data points in the same class (having the same label) will stay relatively closer while data points in different classes (having different labels) will stay relatively farther apart (see [19], [6] and [23]). However, when the weight matrix connecting data points is not completely consistent with the labels of data points, excessive similarity propagation through the graph will harm the classification, therefore, we use label information to guide the similarity-propagation process on the graph. This motivated us to use the label information of training data to optimize the parameter in a random-walk kernel.

A t-step random-walk kernel is generally derived from a transition matrix with a t-step random walk by normalization and symmetrization. Given a base kernel K with positive entries (in this paper, we use profile kernels), the transition matrix P of a one-step random walk is defined as follows: let P_{ij} be the probability $P(x_i \to x_j)$, then after t steps of a random walk, the transition probability can be calculated as $P^t = (D^{-1}K)^t$, where D is a diagonal matrix with $D_{ii} = \sum_k K_{ik}$. Ideally, we want to use P^t as the kernel matrix for SVM classification. However, a kernel matrix

must be a symmetric positive semi-definite matrix, therefore, we do the following manipulations to derive a kernel matrix from P^t. As is described in [23], let $L = D^{-\frac{1}{2}}KD^{-\frac{1}{2}}$, with its eigen-decomposition, $L = U\Lambda U^T$, and $\tilde{L} = U\Lambda^t U^T$, where, t denotes the exponent, and, T, denotes the transpose. Then, the new kernel corresponding to a t-step random walk is calculated as $\tilde{K} = \tilde{D}^{-\frac{1}{2}}\tilde{L}\tilde{D}^{-\frac{1}{2}}$, where $\tilde{D}$ is a diagonal matrix with $\tilde{D}_{ii} = \tilde{L}_{ii}$. We can see that the derived kernel $\tilde{K}$ relates to the transition matrix after t-steps of a random walk P^t as follows: $\tilde{K} = \tilde{D}^{-\frac{1}{2}}D^{\frac{1}{2}}P^tD^{-\frac{1}{2}}\tilde{D}^{-\frac{1}{2}}$.

A random-walk kernel based on PSI-BLAST E-values has been tried in [23] for protein remote homology detection. The challenge in random-walk kernels is how to decide the optimal number of random steps. Since random walks exploit both labeled data and unlabeled data to estimate the manifold structure of data, performing too many steps of a random walk can lead to the possibility of nearby clusters joining together, resulting in data points in different classes come closer. On the other hand, if the number of steps is too small, it can lead to a separation of data points in the same class. Our goal is to find the optimum number of steps that is most consistent with the class memberships of the data points. Using the label information of training data to learn the parameters of kernel functions has been successfully adopted by researchers. Related research can be found in [25], [13] and [16]. Here, we need to learn the parameters of the random-walk kernel that achieves the goal of max-margin classification using the label information of training data. A brute-force solution to this problem results in a non-convex optimization problem, therefore, we propose using a positive linear combination of the base kernel and random-walk kernels from one step to m steps to calculate a new kernel to approximate the kernel with the optimum number of random steps by optimizing the dual objective function of the resulting SVM. We call the resulting kernel "learned random-walk kernel". Since every t-step random-walk kernel has trace n, if the base kernel also has trace n, by restricting the learned kernel to have trace n too, a positive linear combination of the base kernel and the random-walk kernels leads to a convex combination of these kernels, where n is the total number of training data and test data points. The result is the following optimization problem:

$$\begin{aligned} min_\mu max_\alpha \quad & 2\alpha^T\mathbf{1} - \alpha^T(K_{tr}\otimes yy^T)\alpha, \\ s.t. \quad & \alpha^T y = 0 \\ & \mathbf{0} \le \alpha \le C\mathbf{1}, \\ & K = \mu_0\tilde{K}^0 + \textstyle\sum_{k=1}^m \mu_k\tilde{K}^k, \\ & \textstyle\sum_{k=0}^m \mu_k = 1, \\ & \mu_k \ge 0, \qquad k = 0,\ldots,m, \end{aligned} \tag{3.9}$$

where $\tilde{K}^0$ is the base kernel for deriving the learned random-walk kernel, $\tilde{K}^k$ is the random-walk kernel with a k-step random walk, and, m, is the maximal number of random steps performed. The above optimization problem is a special case of the optimization problem discussed in [13]. We follow the framework as is shown in [13], and show that the above problem is equivalent to the following quadratically constrained convex optimization problem:

$$\begin{aligned} min_{\alpha,t} \quad & t, \\ s.t. \quad & t \ge \alpha^T(\tilde{K}^k_{tr}\otimes yy^T)\alpha - 2\alpha^T\mathbf{1}, \quad k = 0,\ldots,m, \\ & \alpha^T y = 0 \\ & \mathbf{0} \le \alpha \le C\mathbf{1}, \end{aligned} \tag{3.10}$$

where tr denotes the training part of the corresponding kernel. The optimal values of parameters $\mu_k, k = 0,\ldots,m$ are exactly the dual solution to the above quadratic constrained convex optimization problem. They can be found using the standard optimization software SeDuMi [18] or MOSEK [2] which solve the primal and dual of an optimization problem simultaneously. For huge datasets, we can use SMO-like gradient-based algorithms [20] to solve the above problem. In this work, all the optimization problems were solved using MOSEK.

THEOREM 3.1. *The optimization problem in equation 3.9 is equivalent to the optimization problem in equation 3.10.*

Proof. It's easy to see that all the constraints in equation 3.9 are linear thus convex with respect to α and μ. Let $\ell = 2\alpha^T\mathbf{1} - \alpha^T(K_{tr}\otimes yy^T)\alpha$, since only K_{tr} appears in ℓ in equation 3.9, K_{tr} is the only part we need from K to solve equation 3.9. ℓ is linear thus convex with respect to μ. The Hessian of ℓ with respect to α is $-(K_{tr}\otimes yy^T)$, which is negative semi-definite, hence, ℓ is concave with respect to α. And ℓ is continuous with respect to α and μ. Therefore, we have the following equations:

$$\begin{aligned} & min_{\mu:\ \mu\ge\mathbf{0},\ \sum_{k=0}^m\mu_k=1}\ max_{\alpha:\ \alpha^Ty=0,\ \mathbf{0}\le\alpha\le C\mathbf{1}} \\ & \qquad 2\alpha^T\mathbf{1} - \alpha^T[(\sum_{k=0}^m \mu_k\tilde{K}^k_{tr})\otimes yy^T]\alpha \\ = & \ max_{\alpha:\ \alpha^Ty=0,\ \mathbf{0}\le\alpha\le C\mathbf{1}}\ min_{\mu:\ \mu\ge\mathbf{0},\ \sum_{k=0}^m\mu_k=1} \\ & \qquad 2\alpha^T\mathbf{1} - \alpha^T[(\sum_{k=0}^m \mu_k\tilde{K}^k_{tr})\otimes yy^T]\alpha \end{aligned}$$

$$
\begin{aligned}
&= max_{\alpha:\ \alpha^T y=0,\ \mathbf{0}\le\alpha\le C\mathbf{1}}\ \ min_{\mu:\ \mu\ge\mathbf{0},\ \sum_{k=0}^{m}\mu_k=1} \\
&\qquad \sum_{k=0}^{m}\mu_k[2\alpha^T\mathbf{1}-\alpha^T(\tilde{K}_{tr}^k\otimes yy^T)\alpha] \\
&= max_{\alpha:\alpha^T y=0,\mathbf{0}\le\alpha\le C\mathbf{1}} min_k[2\alpha^T\mathbf{1}-\alpha^T(\tilde{K}_{tr}^k\otimes yy^T)\alpha] \\
&= max_{\alpha,t:\ \ \alpha^T y=0,\ \ \mathbf{0}\le\alpha\le C\mathbf{1},t\le 2\alpha^T\mathbf{1}-\alpha^T(\tilde{K}_{tr}^k\otimes yy^T)\alpha} \quad t \\
&= min_{\alpha,t:\alpha^T y=0,\mathbf{0}\le\alpha\le C\mathbf{1},\ \ t\ge\alpha^T(\tilde{K}_{tr}^k\otimes yy^T)\alpha-2\alpha^T\mathbf{1}} \quad t
\end{aligned}
\tag{3.11}
$$

The first equality holds due to the special property of ℓ described above according to [5]. The second and third equalities hold due to the properties of the simplex defined by μ. The last two equalities hold due to the rewriting of the optimization problems in different formats. The last equality shows that the optimization problem in equation 3.9 is equivalent to the optimization problem in equation 3.10.

As is described in [10], the ideas of random walks and diffusion are closely related. Given a kernel matrix K, we can view it as a similarity matrix and compute the graph laplacian as $Q = D-K$, where D is a diagonal matrix described in this section. Instead of taking the form of the t-th power of the transition matrix P as in random-walk kernels, a diffusion kernel $K^{diffuse}$ takes a form of the matrix exponential of Q:

$$
\begin{aligned}
K^{diffuse} &= e^{\beta Q} = lim_{n->\infty;n\in\mathcal{N}}\ \ (I+\frac{\beta Q}{n})^n \\
&= I+\beta Q+\frac{\beta^2}{2}Q^2+\ldots+\frac{\beta^t}{t!}Q^t+\cdots \\
&= \sum_i v_i e^{\beta\lambda_i}v_i^T,
\end{aligned}
\tag{3.12}
$$

where β is a real parameter to control the diffusion, which is analogous to the minus inverse squared variance parameter in Gaussian kernels, I is an identity matrix, $\mathcal{N}$ is the integer set, and, v_i and λ_i are the i-th eigenvalue and eigenvector of K respectively. The first line in the above equation can be interpreted as a random walk with an infinite number of infinitesimally small steps. In this paper, we compute diffusion kernels based on profile kernels, and compare their performance to that of learned random-walk kernels shown in the experimental results section.

The computation of both a random-walk kernel and a diffusion kernel requires the eigen-decomposition of a base kernel, which has a worst-case time complexity $O(n^3)$. Computing the learned random-walk kernel described above requires solving in addition, the quadratically constrained convex optimization problem in equation 3.10, which has a worst-case time complexity $O(mn_{tr}^3)$ using an interior-point method, where n_{tr} is the number of training data points.

4 Protein sequence motif discovery using learned random-walk kernels

To identify sequence motifs making important contributions to discriminating the remote homology membership of a protein sequence x, we calculate the j-th positional contribution to the positive classification of sequence x using the following equation:

$$
\sigma(x[j]) = max(\sum_{i=1}^{n_{tr}}\alpha_i K(i, x[j-k+1:j+k-1]), 0),
\tag{4.13}
$$

where i indexes training sequences, $x[j-r+1:j+r-1])$ represents a subsequence window with radius r centered at position j, $K(i, x[j-r+1:j+r-1])$ represents the contribution to the kernel entry $K(i,x)$ made by $x[j-r+1:j+r-1])$, K is the learned random-walk kernel, and α is the dual parameter of the SVM based on K. However, the mapping from the base kernel which is a profile kernel to the learned random-walk kernel is not linear, so there is no closed-form solution to calculate $K(i, x[j-r+1:j+r-1])$. Instead, we resort to the following algorithm to calculate $\hat{K}(i, x[j-r+1:j+r-1])$, which is an approximation to $K(i, x[j-r+1:j+r-1])$.

Algorithm 4.1. The algorithm for computing positional contribution to positive classification $\sigma(x[j])$
Input: sequence profiles P, sequence x, position j, radius r, μ, α, profile kernel matrix K^{prof}, and learned random-walk kernel matrix K.
Output: $\hat{K}(\cdot, x[j-r+1:j+r-1])$ and $\sigma(x[j])$
1. Use P to compute the contribution to the profile-kernel matrix made by $x[j-r+1:j+r-1]$, denoted by M, which is symmetric and has non-zero entries only in the row and the column corresponding to sequence x.

2. Cosinely normalize K^{prof} and M using diagonal entries in K^{prof}. $\tilde{K}_{ij}^0 = \frac{K_{ij}^{prof}}{\sqrt{K_{ii}^{prof}K_{jj}^{prof}}}$, and $\tilde{M}_{ij} = \frac{M_{ij}}{\sqrt{K_{ii}^{prof}K_{jj}^{prof}}}$.

3. Compute new learned random-walk kernel matrix K' based on new base kernel matrix $(\tilde{K}^0-\tilde{M})$ and input combination coefficient μ.

4. $\hat{K}(\cdot, x[j-r+1:j+r-1]) = K(\cdot,x) - K'(\cdot,x)$.

5. Replace K with $\hat{K}$ in equation 4.13 to compute $\sigma(x[j])$.

We can use the above algorithm to compute the positional contribution score to positive classification for

Kernels	Overall Mean ROC_{50}
the best profile kernel	0.87
diffusion kernel	0.79
2-step random-walk profile kernel	0.86
learned random-walk kernel	0.90

Table 1: Overall Mean ROC_{50} scores over the 54 families corresponding to different kernels.

both positive training and test sequences. Then we can rank the positions by the positional contribution score σ, and the top ranked positions, which occupy above 90% of the total positional contribution score mass, can be regarded as essential regions discriminating the remote homology membership of the considered sequences.

5 Experiments

5.1 Experimental results on protein remote homology detection We determine the classification performance of learned random-walk kernel against the best profile kernel and the random-walk kernel with a fixed number of random steps by comparing their ability to detect protein remote homology. We used the benchmark dataset, derived by Jaakkola from the SCOP database for this purpose (see [17] and [9]). In SCOP, protein sequences are classified into a three-level hierarchy: Fold, Super-family, and Family, starting from the top. Remote homology is simulated by choosing all the members of a family as positive test data, some families in the same super-family of the test data as positive training data, all sequences outside the fold of the test data as either negative training data or negative test data, and sequences that are neither in the training set nor in the test set as unlabeled data. This data splitting scheme has been used in several previous papers (see [9], [15], and [23]). We used the same training and test data split as that used in [15] and [23]. We used version 1.59 of the SCOP dataset (http://astral.berkeley.edu), in which no pair of sequences share more than 95% identity.

In the data splits, of most experiments, there are only a few positive test cases but, hundreds, or even thousands of negative test cases. The maximum number of positive test cases is usually below 30, but the maximum number of negative test cases is above 2600. The minimum number of positive test cases is 1, but the minimum number of negative test cases is still above 250. In the experiments with a very limited number of

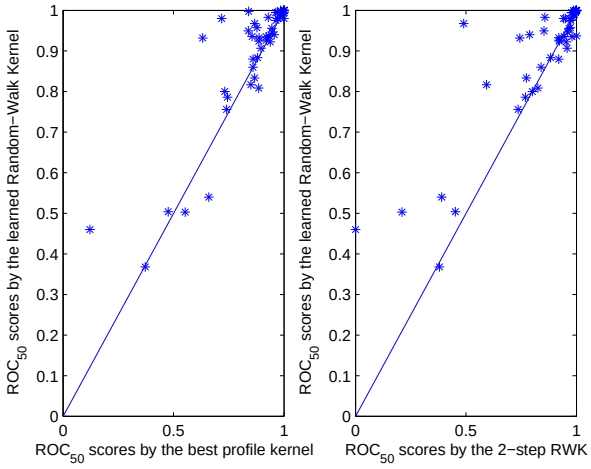

Figure 2: The left figure shows the scatter plot of the ROC_{50} scores produced by the leanred random-walk kernel vs. the ROC_{50} scores produced by the best profile kernel; and the right figure shows the scatter plot of the ROC_{50} scores produced by the leanred random-walk kernel vs. the ROC_{50} scores produced by the best random-walk kernel with a fixed number of random steps.

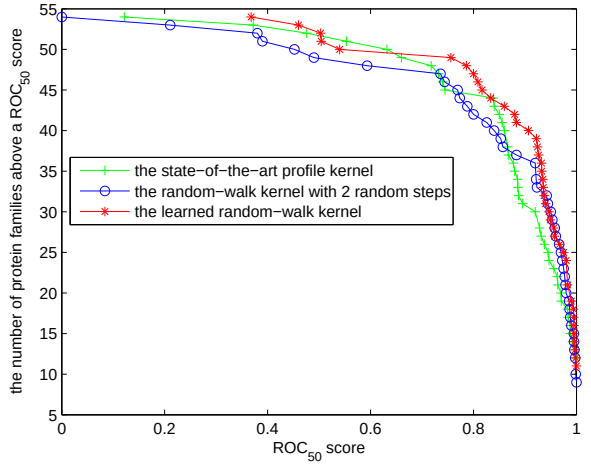

Figure 3: The number of protein families with ROC_{50} scores above a particular value for different kernels using the best profile kernels.

Kernels	ROC_{50} score
the best profile kernel	0.12
2-step random-walk profile kernel	0.00
learned random-walk kernel	0.46

Table 2: The ROC_{50} scores on the most difficult protein family **Glutathione S-transferases, N-terminal domain** corresponding to different kernels.

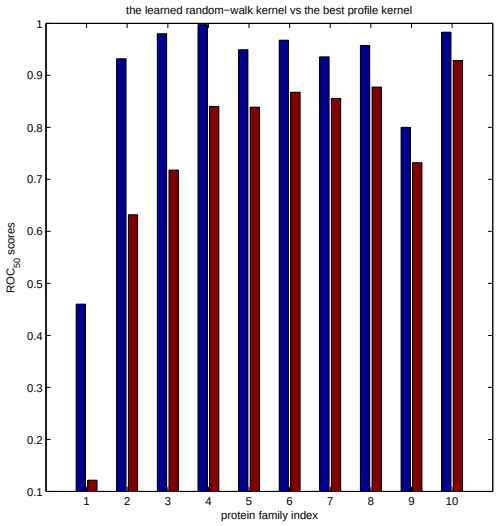

Figure 4: The top 10 largest improvement of ROC_{50} scores given by the learned random-walk kernel over the best profile kernel. The blue bars correspond to the learned random-walk kernel and the red bars correspond to the best profile kernel.

positive test cases and a large number of negative test cases, we can almost ignore the ranking of positive cases below 50 negative cases. In such situations, we consider the ROC_{50} score much more informative of prediction performance of different methods than the ROC score. Here, a ROC curve plots the rate of true positives as a function of the rate of false positives at different decision thresholds. The ROC score is the area under the curve. The ROC_{50} score is the ROC score computed up to the first 50 false positives. Thus, in our experiments, we only compare the ROC_{50} scores corresponding to different kernels (for possible comparison to old results, we also give ROC scores).

Since the optimization procedure for calculating the linear combination coefficients for combining random-walk kernels is highly dependent on labels, we adopted the following approach: prior to training SVM, close homologs of the positive training data in the unlabeled set found by PSI-BLAST with E-value less than 0.05 are added to the positive training set, and are labelled as positive. When training SVM based on random-walk kernels with a fixed number of random steps, we also used unlabeled data as discussed above. In the experiments, the maximum number of steps of random walks m for the learned random-walk kernel is set to be 6 (when it is set to be a number from 7 to 10, the computational time is longer but the results produced are similar to that by $m = 6$.). And we used hard-margin SVM to identify protein remote homology (the free parameter C in SVM is set to be infinity, which has been shown very effective for protein classification [23]).

Table 1 gives the overall mean ROC_{50} scores over 54 protein families for different kernels. From Table 1, we see that the diffusion kernel produces much worse performance than the base kernel. The poor performance of the diffusion kernel is probably due to that the free parameter β chosen by cross validation is not optimal. Table 2 shows the ROC_{50} scores on the most difficult protein family **Glutathione S-transferases, N-terminal domain** on which all the previous approaches produce very poor performance while our approach gives very good performance. Figure 2 shows the scatter plot of the ROC_{50} scores produced by the leanred random-walk kernel vs. the ROC_{50} scores produced by the best profile kernel and the ROC_{50} scores produced by the best random-walk kernel with a fixed number of random steps (2 steps). Figure 3 shows the number of protein families out of 54 families above each possible ROC_{50} threshold for the learned random-walk kernel and the other two kernels (note that they are not ROC_{50} curves but the summaries of all the ROC_{50} scores). Figure 4 shows the top 10 largest improvement of ROC_{50} scores produced by the learned random-walk kernel over the best profile kernel. Figure 5 shows the top 10 largest improvement of ROC_{50} scores produced by the learned random-walk kernel over the best random-walk kernel with a fixed number of random steps. If we use a fixed number of steps of random walks for all the protein families, the random-walk kernel with 2 steps gives the best average ROC_{50} score. Figure 6 shows how the performance of learned random-walk kernel and the m-step random-walk kernel varies with m. It clearly shows that the performance of random-walk kernel with a fixed number of random steps is very sensitive to the step parameter while the learned

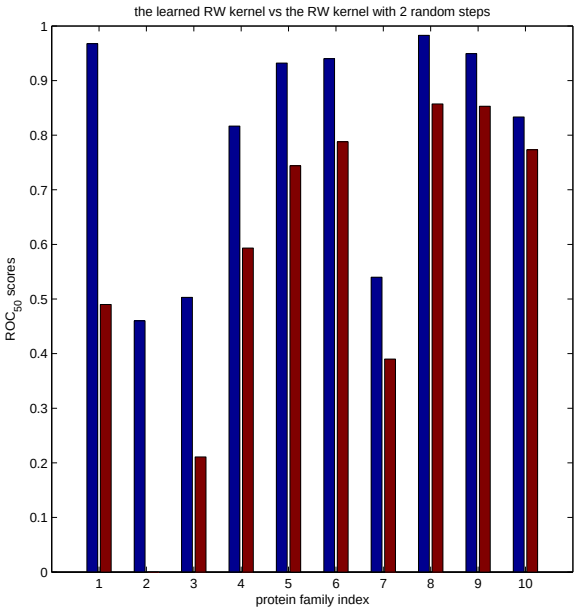

Figure 5: The top 10 largest improvement of ROC_{50} scores given by the learned random-walk kernel over the best random-walk kernel with a fixed number of steps. The blue bars correspond to the learned random-walk kernel and the red bars correspond to the best random-walk kernel with a fixed number of steps.

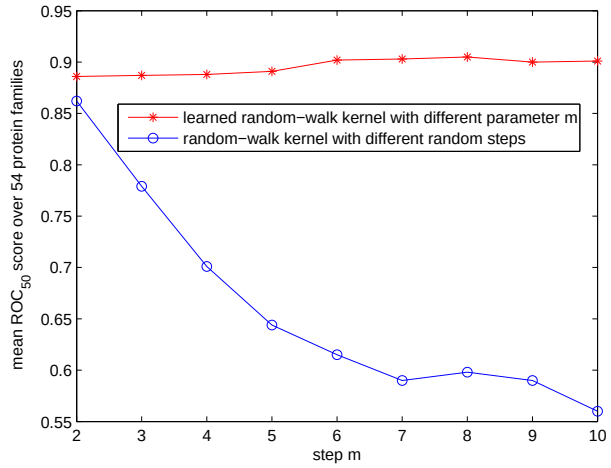

Figure 6: The mean ROC_{50} score over 54 protein families for learned random-walk kernel and k-step random-walk kernel with different choices of m.

random-walk kernel is much more robust. From Table 1 and Figure 2-6, we see that the learned random-walk kernel has much better performance than other kernels (all the methods produce high mean ROC scores over the 54 families; the ROC scores produced by the learned random-walk kernel, 2-step random-walk kernel and the best profile kernel are, respectively, 0.99, 0.97, 0.98).

To determine whether the improvement given by the learned random-walk kernel is statistically significant, we performed Wilcoxon Matched-Pairs Signed-Ranks Tests on the differences between paired kernels. The resulting p-value for the ROC_{50} score difference between the learned random-walk kernel and the best profile kernel is 3.30×10^{-3}, and the p-value for the pair between the learned random-walk kernel and the best profile kernel with 2 steps of random walks is 1.27×10^{-2}. However, the p-value for the ROC_{50} score difference between the best profile kernel with 2 steps of random walks and the best profile kernel is 0.62. Therefore, we can conclude that the improvement given by the learned random-walk kernel is statistically significant.

5.2 Experimental results on protein sequence motif discovery In this subsection, we present the results of motif discovery using the SVMs based on the learned random-walk kernels. We set the radius parameter r in section 4 to 5. Our experimental results show that the important discriminative motifs for a protein sequence often lie in the regions connecting or bordering at common structure motifs such as α-helixes and β-sheets. This completely makes sense in biology. Common structure motifs occur frequently in all kinds of protein sequences, while the regions connecting or bordering at these common motifs represent different ways of assembling these common structures, which should be more important identifiers of remote homology than other regions.

In the following, we will perform a case study for the identification of super-family **ConA-like lectins/glucanases**. On this super-family, the ROC_{50} scores produced by the state-of-the-art profile kernel and the random-walk kernel with 2 random steps are, respectively, 0.63 and 0.74, while the the ROC_{50} score produced by the learned random-walk kernel is 0.93.

Figure 7 shows the distributions of positional contribution scores to the positive classification of 4 positive training sequences with PDB id 1a8d-1, 3btaa1, 1epwa1, and 1kit-2. This figure shows that a small fraction of positions, which are peaky positions in the figure, have much higher scores, meaning that they are much more important than other positions for the identification of remote homology. Figure 8 shows the distribution of the positional contribution scores to positive classifica-

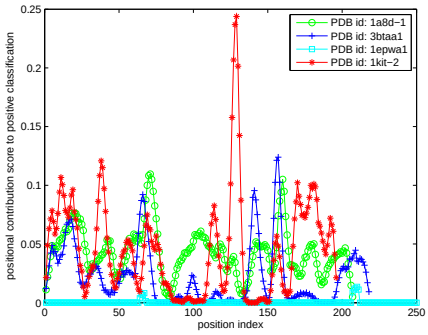

Figure 7: The distributions of positional contribution scores to positive classification for 4 positive training sequences.

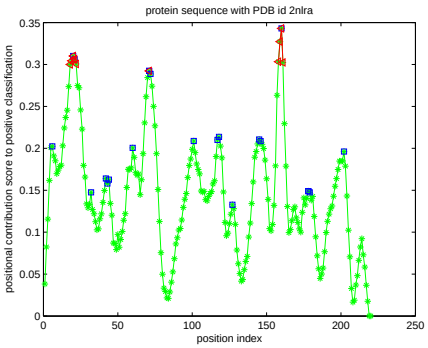

Figure 8: The positional contribution scores to positive classification of a positive training sequence with PDB id 2nlra. The blue positions are local peak positions, and the red positions correspond to the top 10 highest positions.

tion of another positive training sequence with PDB id 2nlra. The blue positions are local peak positions, and the red positions correspond to the top 10 highest positions. Figure 9 shows the motif of sequence 2nlra annotated by PDB, and Figure 10 shows the motif predicted by the SVM based on the learned random-walk kernel, in which the blue and red positions in Figure 8 are also marked blue and red respectively here. From this figure, we can see that the blue and red regions lie either in the center of a standard structure motif, which may represent a standard motif, or lie in the regions connecting or bordering at standard motifs, which may act as bridge motifs.

Figures 11, 12, 13 and 14 show the results for a positive test sequence with PDB id 1c1l. In details, Figure 11 shows the distribution of the positional contribution scores to positive classification of sequence 1c1l, and the red positions correspond to the top 15 highest positions. Figure 12 shows the ROC_{50} scores of predicting the super-family of sequence 1c1l by training SVMs on learned random-walk kernels by respectively removing the subsequence window with radius 5 centered at each position. We can see that the results in Figure 12 are consistent with the positional contribution scores in Figure 11. Figure 13 shows the motif of this sequence annotated by PDB, and Figure 14 shows the motif predicted by the learned random-walk kernel. The red regions correspond to the red positions in Figure 11. Again, we see that the red regions represent standard motifs or act as bridge motifs.

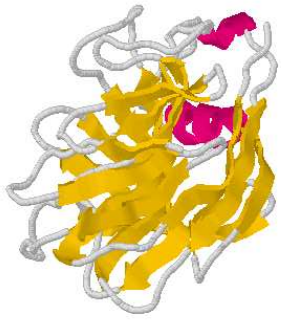

Figure 9: The structure motif annotated by PDB for protein sequence with PDB id 2nlra.

6 Conclusions and discussions

In this paper, we proposed a new approach to approximate the optimal number of steps of random walks

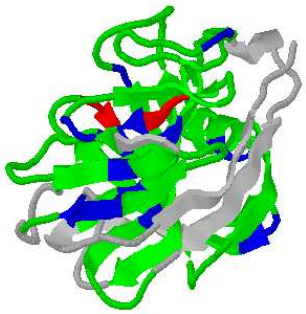

Figure 10: The sequence motif discovered by the SVM based on the learned random-walk kernel for protein sequence with PDB id 2nlra. The sum of the positional contribution scores of the green regions are above 80% of the sum of all the positional scores in 2nlra. The red regions correspond to the top 10 ranked positions, which correspond to the red positions in Figure 8. The blue regions correspond to the blue positions in Figure 8.

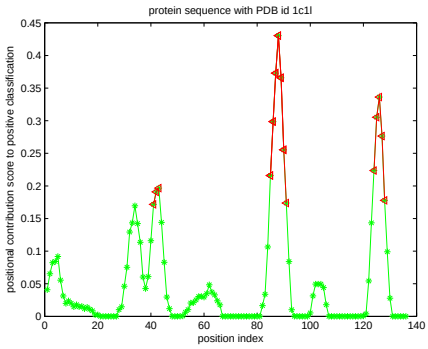

Figure 11: The positional contribution scores to positive classification of a positive test sequence with PDB id 1c1l. The red positions correspond to the top 15 highest positions.

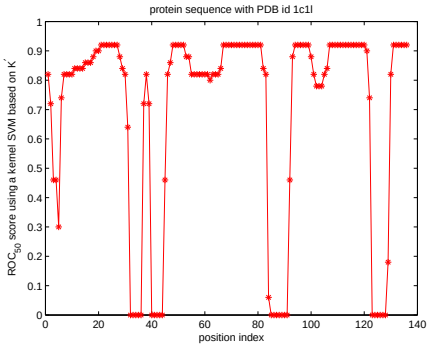

Figure 12: The ROC_{50} score obtained using a kernel SVM based on K' after removing the subsequence window with radius 5 centered at each position for the positive classification of the sequence with PDB id 1c1l.

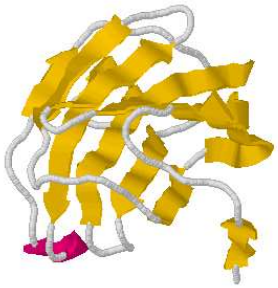

Figure 13: The structure motif annotated by PDB for protein sequence with PDB id 1c1l.

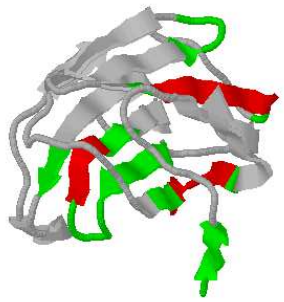

Figure 14: The sequence motif discovered by the SVM based on the learned random-walk kernel for protein sequence with PDB id 1c1l. The sum of the positional contribution scores of the green regions are above 90% of the sum of all the positional scores in 1c1l. And the red regions correspond to the top 15 ranked positions, which are also marked red in Figure 11.

in random-walk kernel by calculating a convex combination of random-walk kernels with different numbers of random steps. The experimental results on protein remote homology detection show that the learned random-walk kernel produces strikingly better performance than previous methods including the best approaches to solving this problem. Out of 54 protein families, the best profile kernel produces ROC_{50} scores above 0.90 on 30 families. This means that the base kernel (the best profile kernel) has already given good results on more than half of the protein families. In contrast, Figure 2 and 3 show that the learned random-walk kernel gives almost 100% correctness on most of the 54 protein families. Moreover, we applied this approach to identify protein sequence motifs that are responsible for discriminating the remote homology of protein sequences. Our results show that the discriminative sequence motifs often represent an important standard structure motif or act as bridge motifs connecting standard structure motifs.

The proposed approach admits a convex optimization problem so there is no concern of local minima. And it makes the performance of random-walk kernels no longer sensitive to the step parameter. Moreover, it is scalable to large datasets.

The proposed approach can successfully solve the discussed biological problem with limited labeled data because it makes use of a large number of pairwise sequence similarities, unlabeled data, and limited labeled data to derive a new kernel. The new kernel corresponds to new similarity metric for pairwise sequences. In the approach, pairwise sequence similarities contribute to defining the transition probability matrix for the random walks, and the convex optimization procedure make the obtained new kernel well reflect the manifold structure of sequences consistent with the labels of training sequences. That is, in the learned random-walk kernel, the kernel entries for pairwise sequences in the same super-family become large and the kernel entries for pairwise sequences in different superfamilies become small.

Our approach here is general and is readily applied to other biological classification problems such as Protein-Protein interaction prediction, Transcription Factor Binding Site prediction and gene function prediction etc. And the learned random-walk kernel here can also be applied to non-biological problems such as document classification, handwritten digit classification and face recognition etc, where we can construct k-step random-walk kernels on texts and images instead of on biological data.

Acknowledgement

We thank MOSEK for providing a free student license. This project was funded by a start-up fund from University of Toronto to Zhaolei Zhang, an NSERC grant to Anthony Bonner, and a grant from Genome Canada through the Ontario Genomics Institute.

References

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

[2] Andersen, E. D. and Andersen, A. D.: The MOSEK interior point optimizer for linear programming: An implementation of the homogeneous algorithm. In Frenk, H., Roos, C., Terlaky, T., and Zhang, S., editors, High Performance Optimization, pages 197-232. Kluwer Academic Publishers. (2000)

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

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

[5] Boyd, S. and Vandenberghe, L.: Convex optimization. Stanford University. (2003)

[6] Chapelle, O., Weston, J., and Schoelkopf., B.: Cluster Kernels for Semi-Supervised Learning. NIPS. (2002)
[7] Cohn, H., Kleinberg, R., Szegedy, B., and Umans, C.: Group-theoretic Algorithms for Matrix Multiplication. Proceedings of the 46th Annual Symposium on Foundations of Computer Science. Pittsburgh, PA, IEEE Computer Society, pp. 379-388. (2005)
[8] Cristianini, N., Shawe-Taylor, J., Elisseeff, A., and Kandola, J.: On kernel-target alignment. In Advances in NIPS. (2001)
[9] Jaakkola, T., Diekhans, M., and Haussler, D.: A discriminative framework for detecting remote protein homologies. Journal of Computational Biology. **7** (2000) Numbers 1/2, 95-114
[10] Kondor, R., and Lafferty, J.: Diffusion kernels on graphs and other discrete structures. In Proceedings of ICML. (2002)
[11] Krogh, A., Brown, M., Mian, I., Sjolander, K., and Haussler, D.: Hidden markov models in computational biology: Applications to protein modeling. Journal of Molecular Biology. **235** (1994) 1501-1531.
[12] Kuang, R., Ie, E., Wang, K., Wang, K., Siddiqi, M., Freund, Y., and Leslie C.: Profile-based String Kernels for Remote Homology Detection and Motif Extraction. Journal of Bioinformatics and Computational Biology. **3** (2005) No. 3 527-550
[13] Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L. and Jordan, M.: Learning the kernel matrix with semidefinite programming. Journal of Machine Learning Research. **5** (2004) 27-72
[14] Leslie, C., Eskin, E., Weston, J., and Noble, W.S.: Mismatch string kernels for SVM protein classification. Neural Information Processing Systems. **15** (2002)
[15] Liao, C. and Noble, W.S.: Combining pairwise sequence similarity and support vector machines for remote protein homology detection. Proceedings of RECOMB. (2002)
[16] Min, R., Bonner, A. and Zhang Z.: Modifying kernels using label information improves SVM classification performance. Proceedings of the International Conference on Machine Learning and Applications. (2007)
[17] Murzin A. G., Brenner S. E., Hubbard T., Chothia C.: SCOP: a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Biol. **247** (1995) 536-540
[18] Sturm, J. F.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optimization Methods and Software, **12**:625-653. Special issue on Interior Point Methods (CD supplement with software). (1999)
[19] Szummer, M. and Jaakkola, T.: Partially labeled classification with Markov random walks. Advances in Neural Information Processing Systems 14. (2001)
[20] Platt, J.C.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. Advances in Kernel Methods - Support Vector Learning, B. Scholkopf, C. Burges, and A. Smola, eds., pp. 185-208, MIT Press. (1999)
[21] Scholkopf, B. and Smola, A.J. : Learning with Kernels. MIT Press, Cambridge, MA. (2002)
[22] Tsuda, K., H. H. Shin and B. Scholkopf: Fast Protein Classification with Multiple Networks. Bioinformatics. **21**(Suppl. 2), 59-65 (09 2005)
[23] Weston, J., Leslie, C., Ie, E., Zhou, D., Elisseeff, A. and Noble, W.S.: Semi-Supervised Protein Classification using Cluster Kernels. Bioinformatics. **21** (2005) 3241-3247.
[24] V. Vapnik: The Nature of Statistical Learning Theory. Springer Verlag, New York. (1995)
[25] Zhu, X., Kandola, J., Ghahramani, Z., and Lafferty, J.: Nonparametric Transforms of Graph Kernels for Semi-Supervised Learning. Advances in Neural Information Processing Systems. **17** (2005)