BIOINFORMATICS

Vol. 20 no. 11 2004, pages 1682–1689
doi:10.1093/bioinformatics/bth141

Protein homology detection using string alignment kernels

Hiroto Saigo[1], Jean-Philippe Vert[2,*], Nobuhisa Ueda[1] and Tatsuya Akutsu[1]

[1]Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji, 611-0011, Japan and [2]Centre de Géostatistique, Ecole des Mines de Paris, 35 rue Saint-Honoré, Fontainebleau, 77300, France

Received on April 30, 2003; revised on December 9, 2003; accepted on January 8, 2004
Advance Access publication February 26, 2004

ABSTRACT

Motivation: Remote homology detection between protein sequences is a central problem in computational biology. Discriminative methods involving support vector machines (SVMs) are currently the most effective methods for the problem of superfamily recognition in the Structural Classification Of Proteins (SCOP) database. The performance of SVMs depends critically on the kernel function used to quantify the similarity between sequences.

Results: We propose new kernels for strings adapted to biological sequences, which we call local alignment kernels. These kernels measure the similarity between two sequences by summing up scores obtained from local alignments with gaps of the sequences. When tested in combination with SVM on their ability to recognize SCOP superfamilies on a benchmark dataset, the new kernels outperform state-of-the-art methods for remote homology detection.

Availability: Software and data available upon request.

Contact: Jean-Philippe.Vert@mines.org

INTRODUCTION

As the number of protein sequences in biochemical databases keeps increasing much faster than our ability to experimentally characterize their functions, the need for accurate protein annotation from an amino acid sequence only is more than ever a central problem in computational biology. A core tool in the annotation process is the detection of sequence similarities, because homology often implies functional similarity. While satisfactory methods exist to detect homologs with a high level of similarity, remote homologs are often difficult to separate from pairs of proteins that share similarities due to chance. Detecting homologs in the so-called 'twilight zone' remains challenging nowadays.

A large panoply of methods and algorithms have been proposed to detect homology between amino acid sequences. Historically, an important milestone in this collection is the Smith–Waterman (SW) algorithm (Smith and Waterman, 1981), which measures the similarity between two sequences by a local gapped alignment. While still widely used in its original form to compare small numbers of sequences, more efficient heuristic algorithms such as BLAST (Altschul *et al*., 1990) or FASTA (Pearson, 1990) have been developed to detect homologies in large databases. Better annotation accuracy was later obtained by comparing a candidate protein with pools of annotated or unannotated proteins with methods such as profiles for protein families (Gribskov *et al*., 1990), hidden Markov models (HMMs) (Krogh *et al*., 1994; Baldi *et al*., 1994), PSI-BLAST (Altschul *et al*., 1997) or SAM-T98 (Karplus *et al*., 1998). These methods are generative, in the sense that they fit a model to a set of proteins with a given annotation, and check how well the model explains a candidate protein in order to annotate it or not.

Further accuracy improvement resulted from the use of discriminative approaches, as opposed to generative approaches. A discriminative approach learns a rule to classify any candidate sequence into a class of proteins by using both sequences known to belong to this class (positive examples) and sequences known to be outside the class (negative examples). Particular attention has been paid to the use of support vector machines (SVMs) in this context that are reported to yield good performance. SVM is an algorithm to learn a discrimination rule from a set of positively and negatively labeled examples, which can then be used to predict the class of any new example. A core component of SVM is the kernel function that measures the similarity between any pair of examples, typically as a dot product between vector representations of the examples. Depending on the choice of the kernel function, a number of variants of SVM can be developed with varying performances. In the case of remote homology detection, several kernels for protein domain sequences have been developed in the recent years. The first attempt resulted in the SVM–Fisher method (Jaakkola *et al*., 2000), where a generative HMM is estimated on a set of proteins, and used to extract a vector representation for each protein sequence (the Fisher score vector). The kernel to measure the similarity between two sequences is then obtained as a product between the corresponding Fisher

*To whom correspondence should be addressed.

score vectors. Another attempt to make a kernel for protein sequences is the SVM-pairwise method (Liao and Noble, 2002), which consists of representing each domain sequence by a vector of pairwise similarities with all domains in the training set (each coordinate of this vector is typically the E-value of the SW score), and taking as a kernel the dot product between these vector representations. The spectrum kernel (Leslie *et al*., 2002) and the mismatch kernel (Leslie *et al*., 2003) measure the similarity between sequences by evaluating the amount of similar short subsequences (typically 4–6 amino acids in length) they share. Tested on a benchmark experiment that consists of recognizing homology at the level of superfamilies in the SCOP database, these SVM-based methods are reported to slightly outperform generative methods.

In this paper, we present a new family of kernels for biological sequences, which we call local alignment kernels. They are motivated by the observation that the SW alignment score between two sequences provides a relevant measure of similarity between protein sequences, which incorporates biological knowledge about protein evolution and whose parameters have been optimized over the years to yield reasonable scoring, at least for close homologs. It is, however, not a valid kernel for strings because it lacks positive definiteness, and therefore cannot be used as such by an SVM. Our first contribution in this paper is to introduce a family of valid kernels that mimic the behavior of the SW score, and to highlight the connection between the SW score and these local alignment (LA) kernels. For practical applications, however, LA kernels suffer from the diagonal dominance problem, i.e. the fact that the kernel value decreases extremely fast with the similarity. SVMs are known not to perform well in such cases, and we therefore propose a modification of these kernels to overcome this issue, which involves taking their logarithm and ensuring that they remain positive definite after this operation by adding a diagonal term. We then end up with a family of kernels for strings, which includes a slight modification of the SW score. When tested on a benchmark experiment to detect remote homologs at the SCOP superfamily level, they are shown to outperform all other state-of-the-art SVM-based methods.

ALGORITHMS

Support vector machine

Our approach to protein classification involves the SVM algorithm, which has been developed by Vapnik *et al*. in the early 1990s (Boser *et al*., 1992). In order to discriminate between members (positive examples) and non-members (negative examples) of a given class of proteins (e.g. a superfamily in the SCOP hierarchy), the SVM learns a classification function from a set of positive examples $\mathcal{X}_+$ and negative examples $\mathcal{X}_-$. The classification function takes the form:

$$f(x) = \sum_{i:x_i \in \mathcal{X}_+} \lambda_i K(x, x_i) - \sum_{i:x_i \in \mathcal{X}_-} \lambda_i K(x, x_i), \qquad (1)$$

where the non-negative weights λ_i are computed during training by maximizing a quadratic objective function, and the function $K(.,.)$ is called a kernel function. Any new sequence x is then predicted to be positive (resp. negative) if the function $f(x)$ is positive (resp. negative). More details about how the weights λ_i are computed and the theory of SVM can be found in Vapnik (1998); Cristianini and Shawe-Taylor (2000) and Schölkopf and Smola (2002).

Any function $K(.,.)$ can be used as a kernel function in (1) as long as it satisfies Mercer's conditions, namely that for any number n and any possible set of distinct sequences $\{x_1, \ldots, x_n\}$, the $n \times n$ Gram matrix defined by $K_{i,j} = K(x_i, x_j)$ be symmetric positive semidefinite. We call such functions Mercer kernels, or simply string kernels below.

The kernel function can be thought of as a measure of similarity between sequences. Different kernels correspond to different notions of similarity, and can lead to discriminative functions with different performance. While typical approaches to design kernels consist of first choosing an appropriate vector representation for sequences, and then taking the inner product between these representations as a kernel for sequences (Jaakkola *et al*., 2000; Leslie *et al*., 2002, 2003; Liao and Noble, 2002), we explore below an alternative path that is to start from a measure of similarity known to be relevant for a given problem, and turn it into a valid kernel.

Local alignment kernel

Starting with basic notations, let $\mathcal{A}$ be a finite set, called the alphabet (the set of 20 amino -acids in our case). A string is a concatenation of letters, and we denote by $\mathcal{X} = \{\epsilon\} \cup \bigcup_{i=1}^{\infty} \mathcal{A}^i$ the set of finite-length strings, where ϵ denotes the empty strings. For any string $x \in \mathcal{X}$, the length of x is denoted by $|x|$. For any two strings x and y, xy denotes the string obtained by concatenation of x and y.

Following the work of Haussler (1999), we define a kernel to detect local alignment between strings by convolving simpler kernels. To this end, we first recall the convolution operation for string kernels. Let K_1 and K_2 be two string kernels, then the convolution kernel $K_1 \star K_2$ is the string kernel defined for any two strings x and y by

$$K_1 \star K_2(x, y) = \sum_{x_1 x_2 = x,\ y_1 y_2 = y} K_1(x_1, y_1) K_2(x_2, y_2).$$

It is known that if K_1 and K_2 are valid string kernels, then $K_1 \star K_2$ is also a valid string kernel (Haussler, 1999). For any kernel K, we denote by $K^{(n)}$ the kernel obtained by n convolutions of K with itself.

Convolution kernels can be useful to compare strings of different lengths that share common parts. For example, Watkins (2000) and Haussler (1999) show that the probability $P(x, y)$ of emitting two strings x and y under a pair HMM model is a valid convolution kernel. We now extend this approach to the definition of convolution kernels that mimic local alignment scoring schemes.

To this end, let us first define three basic string kernels. The first kernel, useful to model the null contribution of substrings before and after a local alignment in the score, is the following constant kernel:

$$\forall (x, y) \in \mathcal{X}^2, \quad K_0(x, y) = 1.$$

Next, the alignment between two residues is quantified by the following string kernel, which is null except when strings are reduced to single letters:

$$K_a^{(\beta)}(x, y) = \begin{cases} 0 & \text{if } |x| \neq 1 \text{ or } |y| \neq 1, \\ \exp[\beta s(x, y)] & \text{otherwise,} \end{cases} \quad (2)$$

where $\beta \geq 0$ is a parameter and $s : \mathcal{A}^2 \to \mathbb{R}$ is a symmetric similarity score. Observe that this is a Mercer kernel only for the values of β, which ensure that the matrix $(\exp(\beta s(a, b)))_{a,b\in\mathcal{A}}$ is positive semidefinite. This is the case whatever $\beta \geq 0$ if and only if the matrix $[s(a, b)]_{a,b\in\mathcal{A}}$ is conditionally positive definite (Berg *et al.*, 1984), which can be checked case by case [this holds in particular if the matrix $(s(a, b))_{a,b\in\mathcal{A}}$ is positive semidefinite].

Finally, to translate the affine penalty gap model, we introduce the following string kernel:

$$K_g^{(\beta)}(x, y) = \exp\{\beta\,[g(|x|) + g(|y|)]\},$$

where $\beta \geq 0$ is a parameter and $g(n)$ is the cost of a gap of length n given by

$$\begin{cases} g(0) = 0 & \text{if } n = 0, \\ g(n) = d + e(n - 1) & \text{if } n \geq 1, \end{cases} \quad (3)$$

where d and e are two parameters called gap opening and extension costs. Observe that this is indeed a valid string kernel, as it can be simply written as a scalar product $K_g^{(\beta)}(x, y) = \Phi_g^{(\beta)}(x) \cdot \Phi_g^{(\beta)}(y)$ between one-dimensional vectors given by $\Phi_g^{(\beta)}(x) = \exp(\beta g(|x|))$.

Now, for any fixed integer $n \geq 1$, let us consider the following string kernel:

$$K_{(n)}^{(\beta)}(x, y) = K_0 \star \left(K_a^{(\beta)} \star K_g^{(\beta)}\right)^{(n-1)} \star K_a^{(\beta)} \star K_0.$$

This kernel quantifies the similarity of two strings x and y based on local alignments of exactly n residues. Indeed, the convolution operation sums up the contributions of all possible decompositions of x and y simultaneously into an initial part (whose similarity is measured by K_0), a succession of n aligned residues (whose similarity is measured by $K_a^{(\beta)}$) possibly separated by $n - 1$ gaps (whose similarity is measured by $K_g^{(\beta)}$), and a terminal part (whose similarity is measured by K_0). For $n = 0$ (no residue aligned), we use the kernel $K_{(0)} = K_0$.

In order to compare two sequences through all possible local alignments, it is necessary to take into account alignments with different numbers n of aligned residues. A simple solution is to sum up the contributions of all kernels $K_{(n)}(\beta)$ for $n \geq 0$, which results in the following local alignment kernel (which we call LA kernel below):

$$K_{\mathrm{LA}}^{(\beta)} = \sum_{i=0}^{\infty} K_{(i)}^{(\beta)}. \quad (4)$$

Note that for any pair of finite-length sequences $(x, y) \in \mathcal{X}^2$ the right-hand term of (4) estimated at (x, y) is a convergent series (because it has only a finite number of non-null terms), so $K_{\mathrm{LA}}^{(\beta)}$ is defined as a pointwise limit of Mercer kernels, and is therefore a Mercer kernel by closure property of the class of Mercer kernel under pointwise limit (Berg *et al.*, 1984).

Link with the SW score

The SW score $\mathrm{SW}(x, y)$ between two sequences x and y is the score of the best local alignment with gaps between the two sequences (Durbin *et al.*, 1998), computed by the SW dynamic programming algorithm (Smith and Waterman, 1981). Let us denote by π a possible local alignment between x and y, defined by a number n of aligned residues, and by the indices $1 \leq i_1 < \cdots < i_n \leq |x|$ and $1 \leq j_1 < \cdots < j_n \leq |y|$ of the aligned residues in x and y, respectively. Let us also denote by $\prod(x, y)$ the set of all possible local alignments between x and y, and by $p(x, y, \pi)$ the score of the local alignment $\pi \in \prod(x, y)$ between x and y, i.e. the sum of the contribution of the aligned residues $\sum_{k=1}^{n} s(x_{i_k}, y_{j_k})$, where $s(.,.)$ is the function used in (2), and of the gap costs defined in (3) when gaps exist between aligned residues in x or y. By definition, the SW score $\mathrm{SW}(x, y)$ between sequences x and y can be written as

$$\mathrm{SW}(x, y) = \max_{\pi\in\Pi(x,y)} p(x, y, \pi). \quad (5)$$

By construction, it is easy to check that the local alignment kernel defined in (4) can be written as follows (Vert *et al.*, 2004):

$$K_{\mathrm{LA}}^{(\beta)}(x, y) = \sum_{\pi\in\Pi(x,y)} \exp\left[\beta\, p(x, y, \pi)\right]. \quad (6)$$

From (5) and (6) it follows that:

$$\lim_{\beta\to+\infty} \frac{1}{\beta} \ln K_{\mathrm{LA}}^{(\beta)}(x, y) = \mathrm{SW}(x, y). \quad (7)$$

These equations clarify the link between the LA kernel (4) and the SW score, and highlight why the SW score is theoretically not a valid kernel. First, the SW score only keeps the contribution of the best local alignment to quantify the similarity between two sequences, instead of summing up the contributions of all possible local alignments like the LA kernel does.

Second, the SW score is the logarithm of this alignment score, and taking the logarithm is usually an operation that does not preserve the property of being positive definite (Berg *et al*., 1984).

Diagonal dominance issue

In many cases of practical interest, the LA kernel defined in (4) suffers from the diagonal dominance issue, namely the fact that $K(x,x)$ is easily orders of magnitudes larger than $K(x,y)$ for two different sequences x and y, even though x and y might share interesting similarities. This is particularly evident for increasing values of the parameter β. In practice, it has been observed that SVMs do not perform well in this situation (Schölkopf *et al*., 2002).

In order to decrease the effect of diagonal dominance, we propose to consider the function:

$$\tilde{K}_{\mathrm{LA}}^{(\beta)}(x,y) = \frac{1}{\beta}\ln K_{\mathrm{LA}}^{(\beta)}(x,y). \tag{8}$$

An obvious problem with this operation is that the logarithm of a Mercer kernel is not a Mercer kernel in general (Berg *et al*., 1984; Schölkopf *et al*., 2002). Even though $\tilde{K}_{\mathrm{LA}}^{(\beta)}$ might not be a valid Mercer kernel, it has the following interesting properties. First, observe that $\tilde{K}_{\mathrm{LA}}^{(\beta)}$ is a monotonically increasing function of $K_{\mathrm{LA}}^{(\beta)}$, and that by (6) the value of $\tilde{K}_{\mathrm{LA}}^{(\beta)}(x,y)$ is always non-negative, because at least one local alignment between x and y has a non-negative score. Observe also by (7) that for large β, $\tilde{K}_{\mathrm{LA}}^{(\beta)}$ behaves like the SW score. Intuitively, $\tilde{K}_{\mathrm{LA}}^{(\beta)}$ is therefore of the order of magnitude of the SW score but includes contribution from all possible local alignments.

Because $\tilde{K}_{\mathrm{LA}}^{(\beta)}$ might not be a positive definite kernel, some care must be taken to ensure that the SVM converges to a large margin discrimination rule during learning. We tested two approaches to make the symmetric funtion $\tilde{K}_{\mathrm{LA}}^{(\beta)}$ positive definite on a given training set of sequences, which we now describe.

The first approach we propose is to add to the diagonal of the training Gram matrix a non-negative constant large enough to make it positive definite. In all experiments presented below we perform this operation by subtracting from the diagonal the smallest negative eigenvalue of the training Gram matrix, if there are negative eigenvalues. The resulting kernel, which we call LA-eig, is equal to $\tilde{K}_{\mathrm{LA}}^{(\beta)}$ except eventually on the diagonal.

We compare this approach to the method proposed by Schölkopf *et al*. (2002), which consists of working with the empirical kernel map. In this case, for a given training set $x_1,\ldots,x_n$ of sequences, each possible sequence x is mapped to the n-dimensional vector $(\tilde{K}_{\mathrm{LA}}^{(\beta)}(x,x_1),\ldots,\tilde{K}_{\mathrm{LA}}^{(\beta)}(x,x_n))^{\mathrm{T}}$. These vector representations are then used to train the SVM and predict the class of new sequences. The corresponding kernel between two sequences x and y, which we call the LA-ekm kernel, is therefore equal to $\sum_{i=1}^{n}\tilde{K}_{\mathrm{LA}}^{(\beta)}(x,x_i)\tilde{K}_{\mathrm{LA}}^{(\beta)}(y,x_i)$.

Implementation

The computation of the kernel $K_{\mathrm{LA}}^{(\beta)}$ [and therefore of $\tilde{K}_{\mathrm{LA}}^{(\beta)}$] can be implemented with a complexity in $O(|x|\cdot|y|)$ using dynamic programming by a slight modification of the SW algorithm. Indeed, it can be checked that the kernel is obtained from the following recursive equations (Vert *et al*., 2004):

- Initialization: for $i = 0,\ldots,|x|$ and $j = 0,\ldots,|y|$:

$$\begin{aligned} M(i,0) &= M(0,j) = 0,\\ X(i,0) &= X(0,j) = 0,\\ Y(i,0) &= Y(0,j) = 0,\\ X_2(i,0) &= X_2(0,j) = 0,\\ Y_2(i,0) &= Y_2(0,j) = 0. \end{aligned}$$

- Dynamic programming equations: for $i = 1,\ldots,|x|$ and $j = 1,\ldots,|y|$:

$$\begin{aligned} M(i,j) &= \exp[\beta s(x_i,y_j)]\,[1 + X(i-1,j-1)\\ &\qquad + Y(i-1,j-1) + M(i-1,j-1)],\\ X(i,j) &= \exp(\beta d)M(i-1,j) + \exp(\beta e)X(i-1,j),\\ Y(i,j) &= \exp(\beta d)\,[M(i,j-1) + X(i,j-1)]\\ &\qquad + \exp(\beta e)Y(i,j-1),\\ X_2(i,j) &= M(i-1,j) + X_2(i-1,j),\\ Y_2(i,j) &= M(i,j-1) + X_2(i,j-1) + Y_2(i,j-1). \end{aligned}$$

- Termination:

$$K_{\mathrm{LA}}^{(\beta)}(x,y) = 1 + X_2(|x|,|y|) + Y_2(|x|,|y|) + M(|x|,|y|).$$

Observe that the SW score is obtained exactly by replacing each addition in these equations by a max operator, and by taking the logarithm of the result.

METHODS

We tested the kernels on their ability to classify protein domains into superfamilies in the Structural Classification of Proteins (SCOP) (Murzin *et al*., 1995) version 1.53. We followed the benchmark procedure presented in (Liao and Noble, 2002). The data consist of 4352 sequences extracted from the Astral database (www.cs.columbia.edu/compbio/svm-pairwise), grouped into families and superfamilies. For each family, the protein domains within the family are considered positive test examples, and protein domains within the superfamily but outside the family are considered positive training examples. This yields 54 families with at least 10 positive training examples and five positive test examples. Negative

examples for the family are 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.

To measure the quality of the methods, we use the receiver operating characteristic (ROC) scores, the ROC50 scores, and the median rate of false positives (RFP). The ROC score is the normalized area under a curve that plots true positives against false positives for different possible thresholds for classification (Gribskov and Robinson, 1996). The ROC50 is the area under the ROC curve up to 50 false positives, and is considered a useful measure of performance for real-world application. The median RFP is the number of false positives scoring as high or better than the median-scoring true positives.

All methods involving SVM are tested with a common procedure. We use the Gist publicly available SVM software implementation (http://microarray.cpmc.columbia.edu/gist), which implements the soft margin optimization algorithm described in Jaakkola *et al*. (2000). For any given positive semi-definite kernel Gram matrix 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, i.e. consider the following kernel:

$$K_{\mathrm{norm}}(x, y) = \frac{K(x, y)}{\sqrt{K(x, x)K(y, y)}} + 1. \qquad (9)$$

Second, because many classification problems below are very unbalanced we use a class-dependent regularization parameter that consists of adding to the diagonal a constant $0.02\alpha_+$ (resp. $0.02\alpha_-$) to all positive (resp. negative) examples, where α_+ (resp. α_-) is the fraction of positive (resp. negative) examples in the training set [see Liao and Noble (2002) and Jaakkola *et al*. (2000) for details and justifications].

The LA kernels have several parameters: the gap penalty parameters e and d, the amino acid mutation matrix s and the factor β that controls the influence of suboptimal alignments in the kernel value. A precise analysis of the effects of these parameters would be beyond the scope of this paper so we limit ourselves to the analysis of the effect of the β parameter. In order to be consistent with the SVM-pairwise method, the substitution matrix is always the BLOSUM 62 matrix and the gap parameters are always set to $(e = 11,\ d = 1)$.

For comparison purpose we also tested three other state-of-the-art kernels: the Fisher kernel (Jaakkola *et al*., 2000), the pairwise kernel (Liao and Noble, 2002) and the mismatch kernel (Leslie *et al*., 2003). In each case, we tested the best method presented in the references.

RESULTS

Table 1 summarizes the performance of the various methods. We tested the local alignment kernels LA-eig and LA-ekm for several values of β ranging from $+\infty$, in which case they are derived from the SW score (8), to $\beta = 0.1$.

Table 1. ROC, ROC50 and median RFP averaged over 54 families for different kernels

Kernel	Mean ROC	Mean ROC50	Mean mRFP
LA-eig ($\beta = +\infty$)	0.908	0.591	0.0654
LA-eig ($\beta = 1$)	0.912	0.612	0.0626
LA-eig ($\beta = 0.8$)	0.908	0.597	0.0679
LA-eig ($\beta = 0.5$)	**0.925**	**0.649**	0.0541
LA-eig ($\beta = 0.2$)	0.923	**0.661**	0.0637
LA-eig ($\beta = 0.1$)	0.868	0.429	0.111
LA-ekm ($\beta = +\infty$)	0.916	0.585	0.0580
LA-ekm ($\beta = 1$)	0.920	0.587	**0.0539**
LA-ekm ($\beta = 0.8$)	0.916	0.585	0.0592
LA-ekm ($\beta = 0.5$)	**0.929**	0.600	**0.0515**
LA-ekm ($\beta = 0.2$)	0.877	0.453	0.125
LA-ekm ($\beta = 0.1$)	0.596	0.052	0.500
Pairwise	0.896	0.464	0.0837
Mismatch	0.872	0.400	0.0837
Fisher	0.773	0.250	0.204

The LA-eig and LA-ekm kernels with $\beta = +\infty$ correspond to the SW score (modified to become positive definite on the set of proteins used to train the SVM). Bold numbers indicate the best results in each column.

These results show that both LA-eig and LA-ekm perform best for β in the range 0.2–0.5, and have almost similar performances. This suggests first that the normalization of $\tilde{K}_{\mathrm{LA}}^{(\beta)}$ into a positive definite kernel through the empirical kernel map of (Schölkopf *et al*., 2002) or by subtracting the smallest negative eigenvalue from the diagonal has little influence on the final performance.

Second, the fact that the performance of the LA-eig and LA-ekm kernels is better for β in the range 0.2–0.5 than for $\beta = \infty$ shows that in the context of this paper, the SW score as a kernel is outperformed by variants that take into account suboptimal alignments to quantify the similarity between protein sequences.

More importantly, these experiments show that most of the local alignment kernels tested slightly outperform all three other methods in this benchmark. As an illustration, the distribution of ROC, ROC50 and median RFP scores for all three methods and the LA-eig kernel with $\beta = 0.5$ and $\beta = \infty$ are shown in Figures 1–3. The LA-eig kernel with $\beta = 0.5$ retrieves more than twice as many families as the best other method tested (the pairwise method) at a ROC50 score of 0.8 or higher. This remains true for a wide range of values for β, including $\beta = +\infty$. This means that the SW score as a kernel also outperforms the Fisher, pairwise and mismatch kernels.

Another important factor for practical use of these kernels is their computation cost and speed. The mismatch kernel has a complexity $O(|x| + |y|)$ with respect to the length of the sequences, and is by far the fastest to operate. The complexity of the LA kernel is $O(|x||y|)$. Moreover, the LA kernel is faster to compute for $\beta = +\infty$ (SW score) than for other values of β, because in that case all computations can be

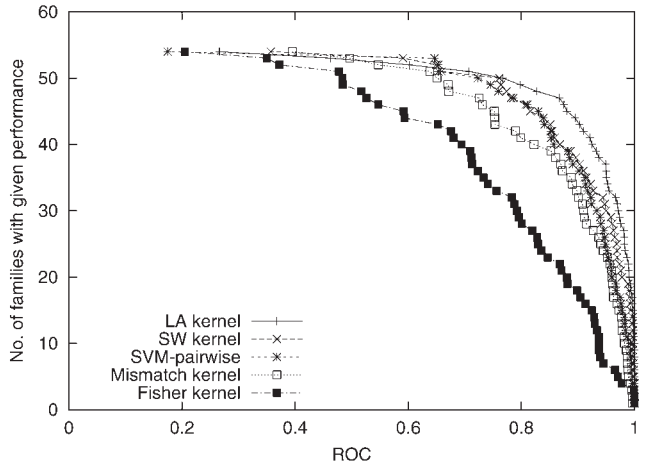

Fig. 1. ROC score distribution for different kernels. The curve denoted LA kernel corresponds to the LA-eig kernel with $\beta = 0.5$. The curve denoted SW kernel corresponds to the LA-eig kernel with $\beta = \infty$, which is equal to the SW score up to a constant on the diagonal.

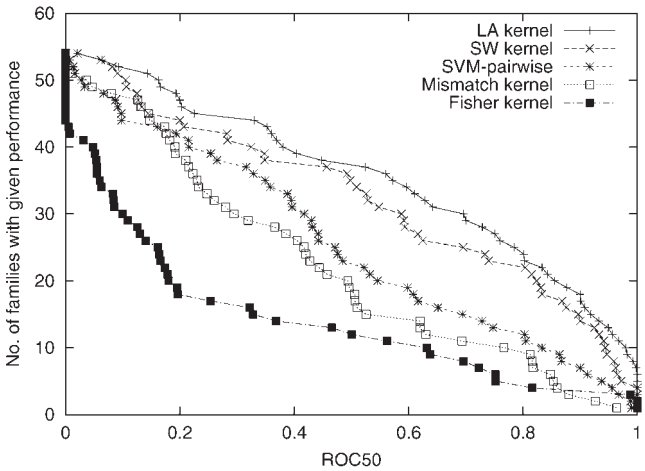

Fig. 2. ROC50 score distribution for different kernels.

performed with sum and max operations on the integer instead of logarithms on floating point real numbers. In our experiments, the computation of the SW kernel was four times slower than the computation of the mismatch kernel (using the SSEARCH software for the SW score, and an implementation of the mismatch kernel described in Leslie *et al*. (2003), which is likely to be optimized in the future.

The computation of the kernel Gram matrix on the training set for SVM-pairwise and the LA-ekm kernels requires $O(n^3)$ further operations to multiply the empirical kernel map matrix by its transpose. Only $O(n^2)$ are approximately required by the LA-eig kernels to compute the smallest eigenvalue using the power method (Golub and Loan, 1996) and subtract it from the diagonal. However, in both cases this operation is

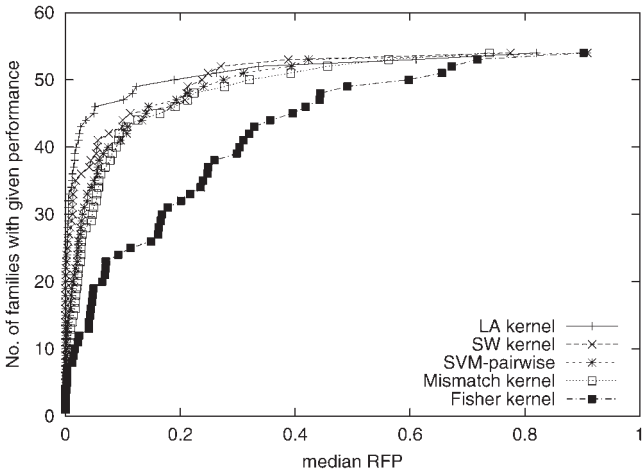

Fig. 3. Median RFP distribution for different kernels.

very fast compared with the time required to compute the LA kernel values or E-values.

Finally, classification of a new sequence with SVM-pairwise or LA-ekm kernels requires computing the explicit empirical kernel map representation of the sequence, i.e. computing n E-value or LA kernels. In the case of the mismatch and LA-eig kernels, the kernels are only computed between the new sequence and the support vector sequences, which usually form only a subset of the training set. Because the performances of LA-eig and LA-ekm are very similar, this suggests to prefer the former to the latter.

DISCUSSION AND CONCLUSION

This paper introduces a family of kernels for protein sequences, based on the detection of high-scoring local alignments. These kernels are biologically motivated, and extend classical work on local alignment scoring to the framework of kernel functions. The theoretically valid local alignment kernels we introduce suffer in practice from diagonal dominance. Hence, we employed a trick to turn them into useful kernels by taking a logarithm and adding a constant on the diagonal. The resulting kernels significantly outperform all other state-of-the-art methods when used with an SVM on a benchmark experiment of SCOP superfamily recognition, which was designed to simulate the problem of remote homology detection. As a result, we obtain a new powerful method for remote protein homology detection.

The remarkable accuracy of our method comes from the combination of two widely used algorithms. On the one hand, the SVM algorithm is based on a sound mathematical framework and has been shown to perform very well on many real-world applications. One of its particularities is that it can perform classification of any kind of data, such as strings in our case, as soon as a kernel function is provided. On the other hand, local alignment scores, in particular the

SW score, have been developed to quantify the similarity of biological sequences. Their parameters have been optimized over the years to provide relevant measures of similarity for homologous sequences, and they now represent core tools in computational biology.

However, direct pairwise comparisons of sequences through local alignment scores such as the SW scores are often considered naive and weak methods to detect remote homology. They are usually outperformed by methods such as PSI-BLAST that extend pairwise comparisons to pools of sequences extracted iteratively. The main contribution of this paper is to show that pairwise sequence comparison can be extremely powerful when used as a kernel function combined with an SVM, and that the SW score itself provides a state-of-the-art method for remote homology detection when used as a kernel. An interesting conclusion of our experiments is that the SW score, however, is outperformed by local alignment scores that sum up the contributions of all possible local alignments. Summing up over local alignments has an important cost in terms of computation time due to the operations required with floating point numbers, but can be worth the cost when one is interested in precision more than in speed. On the other hand, the SW score itself is computed by dynamic programming and is therefore slower to compute than the mismatch kernel that it outperforms. Here again a trade-off must be found between speed and accuracy, depending on the application. However, due to its wide use in computational biology the SW score has been precomputed and stored in databases such as KEGG's SSDB (Kanehisa *et al*., 2002) for virtually all known or predicted proteins of sequenced genomes, which suggests that practical applications for the SW kernel could be implemented in relation with such databases.

The only parameter whose influence was tested is the parameter β of the LA kernel, which controls the importance of the contribution of non-optimal local alignments in the final score. It should be pointed out here that the optimal values for β we observed (in the range 0.2–0.5) are only optimal for an average performance on the 54 families tested, and that the optimal value for each family might fluctuate. Moreover, a number of other parameters could be modified, in particular the gap penalty parameters and the similarity matrix between amino acids, and the optimal values for β might also depend on these parameters. Further theoretical and practical studies, which are beyond the scope of this paper, should be performed to evaluate the influence of these parameters. In particular, it would be interesting to know for which values of these parameters the SW score itself is a valid kernel, and which parameter tuning results in the most accurate remote protein homology detection.

An important open problem with the LA kernels as well as with most other string kernels is the following: how to make the kernel independent of the lengths of the sequences compared? Indeed, long sequences typically result in small kernel values when the kernel is normalized with (9). While much work has been done to estimate the significance of alignment scores for varying sequence length, these approaches remain difficult to adapt to the kernel framework. The importance of this issue might be underestimated in the benchmark experiment presented in this paper, because protein sequences in a SCOP family tend to have similar sequence lengths. However applying kernel-based homology detection in a more realistic setting might reveal important effects of this issue.

Finally, it should be pointed out that possible uses of string kernels go far beyond the single goal of remote homology detection. In combination with SVM or other kernel methods, they can be applied to a variety of problems such as gene structure and function prediction, or heterogeneous data integration, as highlighted in Schölkopf *et al*. (2004).

ACKNOWLEDGEMENTS

We thank Li Liao, William Stafford Noble, Mark Diekhans, Christina Leslie, Eleazar Eskin and Jason Weston for providing information, data and codes related to their work. The computational resource was provided by the Bioinformatics Center, Institute for Chemical Research, Kyoto University, and the Supercomputer Laboratory, Kyoto University. Part of this work was supported by a Grant-in-Aid for Scientific Research on Priority Areas (C) 'Genome Information Science' from MEXT of Japan.

REFERENCES

Altschul,S., Gish,W., Miller,W., Myers,E. and Lipman,D. (1990) A basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

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

Baldi,P., Chauvin,Y., Hunkapiller,T. and McClure,M. (1994) Hidden Markov models of biological primary sequence information. *Proc. Natl Acad. Sci., USA*, **91**, 1053–1063.

Berg,C., Christensen,J. and Ressel,P. (1984) *Harmonic Analysis on Semigroups*. Springer-Verlag, New York.

Boser,B.E., Guyon,I.M. and Vapnik,V.N. (1992) A training algorithm for optimal margin classifiers. *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*. ACM Press, pp. 144–152.

Cristianini,N. and Shawe-Taylor,J. (2000) *An introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Cambridge.

Durbin,R., Eddy,S., Krogh,A. and Mitchison,G. (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge.

Golub,G.H. and Loan,C.F.V. (1996) *Matrix Computations*. John Hopkins University Press.

Gribskov,M., Lüthy,R. and Eisenberg,D. (1990) Profile analysis. *Meth. Enzymol.*, **183**, 146–159.

Gribskov,M. and Robinson,N. (1996) Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Comput Chem.*, **20**, 25–33.

Haussler,D. (1999) Convolution kernels on discrete structures. *Technical Report*. University of California, Santa Cruz.

Jaakkola,T., Diekhans,M. and Haussler,D. (2000) A discriminative framework for detecting remote protein homologies. *J. Comput. Bio.*, **7**, 95–114.

Kanehisa,M., Goto,S., Kawashima,S. and Nakaya,A. (2002) The KEGG databases at GenomeNet. *Nucleic Acids Res.*, **30**, 42–46.

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

Krogh,A., Brown,M., Mian,I., Sjolander,K. and Haussler,D. (1994) Hidden Markov models in computational biology: applications to protein modeling. *J. Mol. Biol.*, **235**, 1501–1531.

Leslie,C., Eskin,E. and Noble,W.S. (2002) The spectrum kernel: a string kernel for SVM protein classification. In Altman,R.B., Dunker,A.K., Hunter,L., Lauerdale,K. and Klein,T.E. (eds), *Proceedings of the Pacific Symposium on Biocomputing 2002*. World Scientific, pp. 564–575.

Leslie,C., Eskin,E., Weston,J. and Noble,W.S. (2003) Mismatch string kernels for SVM protein classification. In Becker,S., Thrun,S. and Obermayer,K. (eds), *Advances in Neural Information Processing Systems 15*. MIT Press, Cambridge.

Liao,L. and Noble,W.S. (2002) Combining pairwise sequence similarity and support vector machines for remote protein homology detection. *Proceedings of the Sixth International Conference on Computational Molecular Biology*. ACM Press, pp. 225–232.

Murzin,A., Brenner,S., Hubbard,T. and Chothia,C. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, **247**, 536–540.

Pearson,W. (1990) Rapid and sensitive sequence comparisons with FASTP and FASTA. *Meth. Enzymol.*, **183**, 63–98.

Schölkopf,B. and Smola,A.J. (2002) *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA.

Schölkopf,B., Tsuda,K. and Vert,J.-P. (2004) *Kernel Methods in Computational Biology*.MIT Press, Cambridge.

Schölkopf,B., Weston,J., Eskin,E., Leslie,C. and Noble,W.S. (2002) A kernel approach for learning from almost orthogonal patterns. In Elomaa,T., Mannila,H. and Toivonen,H. (eds), *Proceedings of ECML 2002, 13th European Conference on Machine Learning*, Helsinki, Finland, August 19–23, Lecture Notes in Computer Science, **2430**. Springer, pp. 511–528.

Smith,T. and Waterman,M. (1981) Identification of common molecular subsequences. *J. Mol. Bio.*, **147**, 195–197.

Vapnik,V.N. (1998) *Statistical Learning Theory*. Wiley, New York.

Vert,J.-P., Saigo,H. and Akutsu,T. (2004) Convolution and local alignment kernels. In Schölkopf,B., Tsuda,K. and Vert,J.-P. (eds), *Kernel Methods in Computational Biology*. MIT Press, pp. 131–154.

Watkins,C. (2000) Dynamic alignment kernels. In Smola,A., Bartlett,P., Schölkopf,B. and Schuurmans,D. (eds), *Advances in Large Margin Classifiers*. Cambridge, MA, MIT Press, pp. 39–50.