BIOINFORMATICS

Vol. 21 Suppl. 1 2005, pages i369–i377
doi:10.1093/bioinformatics/bti1053

RASE: recognition of alternatively spliced exons in C.elegans

G. Rätsch[1,], S. Sonnenburg[2] and B. Schölkopf[3]*

[1]Friedrich Miescher Laboratory of the Max Planck Society, Max Planck, Spemannstrasse 35, Tübingen, Germany, [2]Fraunhofer Institute FIRST, Kekuléstrasse 7, Berlin, Germany and [3]Max Planck Institute for Biological Cybernetics, Spemannstrasse 38, Tübingen, Germany

Received on January 15, 2005; accepted on March 27, 2005

ABSTRACT

Motivation: Eukaryotic pre-mRNAs are spliced to form mature mRNA. Pre-mRNA alternative splicing greatly increases the complexity of gene expression. Estimates show that more than half of the human genes and at least one-third of the genes of less complex organisms, such as nematodes or flies, are alternatively spliced. In this work, we consider one major form of alternative splicing, namely the exclusion of exons from the transcript. It has been shown that alternatively spliced exons have certain properties that distinguish them from constitutively spliced exons. Although most recent computational studies on alternative splicing apply only to exons which are conserved among two species, our method only uses information that is available to the splicing machinery, i.e. the DNA sequence itself. We employ advanced machine learning techniques in order to answer the following two questions: (1) Is a certain exon alternatively spliced? (2) How can we identify yet unidentified exons within known introns?

Results: We designed a support vector machine (SVM) kernel well suited for the task of classifying sequences with motifs having positional preferences. In order to solve the task (1), we combine the kernel with additional local sequence information, such as lengths of the exon and the flanking introns. The resulting SVM-based classifier achieves a true positive rate of 48.5% at a false positive rate of 1%. By scanning over single EST confirmed exons we identified 215 potential alternatively spliced exons. For 10 randomly selected such exons we successfully performed biological verification experiments and confirmed three novel alternatively spliced exons. To answer question (2), we additionally used SVM-based predictions to recognize acceptor and donor splice sites. Combined with the above mentioned features we were able to identify 85.2% of skipped exons within known introns at a false positive rate of 1%.

Availability: Datasets, model selection results, our predictions and additional experimental results are available at http://www.fml.tuebingen.mpg.de/~raetsch/RASE

*To whom correspondence should be addressed.

Contact: Gunnar.Raetsch@tuebingen.mpg.de

Supplementary information: http://www.fml.tuebingen.mpg.de/~raetsch/RASE

1 INTRODUCTION

Alternative splicing is a process through which one gene can generate several distinct proteins or mRNAs. It occurs by alternative usage of exons or parts of exons in pre-mRNA transcripts, and can be specific to a tissue, developmental stage or a condition, such as stress (Maniatis and Tasic, 2002).

Although traditional methods for computational recognition of alternative splicing are usually solely based on expressed sequences (ESTs or cDNAs; cf. Gupta *et al*., 2004 and references therein), more recent techniques tried to identify and exploit local sequence features for prediction (Sorek and Ast, 2003; Sakai and Maruyama, 2004; Dror *et al*., 2004; Sorek *et al*., 2004; Hiller *et al*., 2004). For instance, in Dror *et al*. (2004) features like the exon length, its divisibility by three, the length of the flanking introns and the intensity of the polypyrimidine tract were utilized. Moreover, conservation patterns to another organism have been taken into account. These are among the most discriminative features (Sorek and Ast, 2003). However, this is only possible for a fraction of exons in human (Sorek and Ast, 2003), as exons are frequently not conserved, making the conservational features unavailable. In this work, we aim to design a classifier that accurately distinguishes constitutive from alternatively spliced exons and only uses information that is always available and might also be used by the cellular splicing machinery; i.e. features derived from the exon and intron lengths and features based on the pre-mRNA sequence.

We propose two algorithms for the identification of alternatively spliced exons based on confirmed exons and introns. In the first approach, we propose to use an appropriately designed support vector kernel that is able to deal with DNA sequences (Section 2.2) in order to learn about sequence features near the 3′ and 5′ ends of alternatively spliced exons. The aim is to classify known exons into alternatively and constitutively spliced exons (Section 2.3). It can be applied, for instance, to already

EST confirmed or predicted exons (e.g. GenScan; Burge and Karlin, 1997). However, if we want to apply the method, e.g. to EST confirmed regions, the likelihood is high that an alternatively spliced exon is skipped in the existing sequencing results and was not found by a gene prediction program. We, therefore, propose a second algorithm in Section 2.4 that not only classifies whether a certain exon is alternatively spliced, but it also locates it accurately within an intron. This algorithm can be applied to scan over all EST confirmed introns for skipped exons.

In Section 3, we perform an evaluation of our methods including a biological verification experiment. Moreover, we use novel machine learning techniques in order to understand how the SVM achieves the high accuracy.

2 METHODS

2.1 A database of alternatively and constitutively spliced exons for *Caenorhabditis elegans*

We collected all known *C.elegans* ESTs and cDNAs from Wormbase (Harris *et al*., 2004) (release WS135), dbEST (Boguski *et al*., 1993; as of December 17, 2004) and UniGene (Wheeler *et al*., 2003; as of December 17, 2004). We merged the databases and removed duplicate EST sequences (of either orientation). Using blat (Kent, 2002) we aligned them against the genomic DNA (release WS135). We considered only the sequences with at least 90% sequence identity (over the full length of the sequence). We refined the alignment by correcting typical sequencing errors and by handling polycistronic sequences (see Supplementary website for more details). The alignment was used to confirm exons and introns. Finally, we merged the alignments, if they did not disagree and shared at least one complete exon or intron. For each determined exon and intron, we counted how often they were confirmed by (unique) ESTs.

In the following step we identified pairs of sequences in our set that share the same $3'$ and $5'$ boundaries of the upstream and downstream exon, respectively, where one sequence contains an internal exon and the other does not (i.e. shows evidence of alternative exon usage with the same flanking exon boundaries). This way, we identified 487 exons for which ESTs show evidence for alternative splicing. As negative examples we only considered exon triples that did not show evidence for alternative splicing. We considered this as sufficiently likely when the internal exon and the flanking introns were confirmed by at least two different EST sequences. We were able to extract 2531 exon triples with the internal exon likely to be constitutively spliced. This database of in total 3018 examples is used for training, model selection and evaluation of our methods. The dataset is available on the Supplementary website.

2.2 The weighted degree (WD) kernel

Our approach comprises the use of a discriminative method building on support vector machines (SVMs) (Vapnik, 1995). SVMs construct linear decision rules in a Hilbert space associated with a kernel function (aka similarity measure) k. Specifically, if and only if k satisfies the condition of positive definiteness (Vapnik, 1995), there exists a map Φ into a Hilbert space $\mathcal{H}$ such that $k(\mathbf{s}, \mathbf{s}') = \langle \Phi(\mathbf{s}), \Phi(\mathbf{s}') \rangle$, where $\langle ., . \rangle$ denotes the dot product of $\mathcal{H}$ (Vapnik, 1995). In $\mathcal{H}$, SVMs construct a linear decision rule with large classification margin (Vapnik, 1995). As it determines the map Φ and the space $\mathcal{H}$, the choice of k is crucial when applying an SVM to a given task, and k should take into account the structure of the data and the task as far as possible.

In the present application, the kernel needs to compute similarities between pairs of DNA sequences. Kernels for such tasks have been pioneered by Haussler (1999), Watkins (2000) and Leslie *et al*. (2002) (for further developments, cf. Chapters 4–6 of Schölkopf *et al*., 2004). The present work builds on a particular position-dependent kernel on DNA strings (aka string kernels), the so-called WD kernel (Rätsch and Sonnenburg, 2003), introduces an efficient scheme for computing it, and extends the kernel for the purpose of recognizing alternatively spliced exons. The main idea of the WD kernel is to count the (exact) cooccurrences of $k$mers at corresponding positions in the two sequences to be compared. The WD kernel of order d compares two sequences $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ of equal length L by summing all contributions of $k$mer matches of lengths $k \in \{1, \ldots, d\}$, weighted by coefficients β_k:

$$k(\mathbf{s}_i, \mathbf{s}_j) = \sum_{k=1}^{d} \beta_k \sum_{l=1}^{L-k+1} \mathbf{I}(\boldsymbol{u}_{k,l}(\mathbf{s}_i) = \boldsymbol{u}_{k,l}(\mathbf{s}_j)). \qquad (1)$$

Here, $\mathbf{I}(\text{true}) := 1$ and 0 otherwise, and $\boldsymbol{u}_{k,l}(\mathbf{s})$ is the oligomer of length k starting at position l of the sequence $\mathbf{s}$. The weighting coefficients are fixed in our study to $\beta_k = 2(d - k + 1)/(d(d + 1))$ (Sonnenburg *et al*., 2005 for an algorithm to adaptively determine the βs using multiple kernel learning; MKL). Matching substrings are thus rewarded with a score depending on the length of the substring. Note that although in our case $\beta_{k+1} < \beta_k$, longer matches nevertheless contribute more strongly than shorter ones: this is due to the fact that each long match also implies several short matches, adding to the value of Equation (1). This observation can be used to speed up the kernel computation: as shown in Figure 1, one can identify maximal blocks of agreement in the two sequences, and reward each block depending on its length. The reward r_b of a block of length $b \leq d$ subsumes the weights β_k of all sub-blocks contained in it ($k \leq b$), taking into account that a block of length b contains $b - k + 1$ sub-blocks of length k.[1]

The WD kernel works well for problems where the position of motifs are approximately constant in the sequence or when sufficiently many training examples are available,

[1] A short calculation yields $r_b = (b(3db + 3d - b^2 + 1)/3d(d + 1)$ for $b \leq d$ and $r_b = (b - d)/(3 - 1/3)$ for $b > d$.

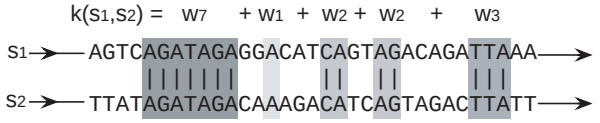

Fig. 1. Given two sequences $\mathbf{s}_1$ and $\mathbf{s}_2$ of equal length, our kernel consists of a weighted sum to which each match in the sequences makes a contribution r_b depending on its length b, where longer matches contribute more significantly.

as it is the case for splice site detection, where the splice factor binding sites appear almost at the same positions relative to the splice site. However, if for instance the sequence was shifted by only 1 nt (Fig. 1), then potentially existing matches would not be found anymore. We, therefore, extend the WD kernel in order to find sequence motifs which are less precisely localized. Our proposed kernel lies in between the completely position-dependent WD kernel and kernels like the so-called spectrum kernel (Leslie *et al*., 2002) which does not use positional information.

The kernel with shifts is defined as

$$k(\mathbf{s}_i, \mathbf{s}_j) = \sum_{k=1}^{d} \beta_k \sum_{l=1}^{L-k+1} \gamma_l \sum_{\substack{s=0 \\ s+l\le L}}^{S(l)} \delta_s \mu_{k,l,s,\mathbf{s}_i,\mathbf{s}_j},$$

$$\mu_{k,l,s,\mathbf{s}_i,\mathbf{s}_j} = \mathbf{I}(\boldsymbol{u}_{k,l+s}(\mathbf{s}_i) = \boldsymbol{u}_{k,l}(\mathbf{s}_j)) + \mathbf{I}(\boldsymbol{u}_{k,l}(\mathbf{s}_i) = \boldsymbol{u}_{k,l+s}(\mathbf{s}_j)), \quad (2)$$

where β_j is as before, γ_l is a weighting over the position in the sequence, $\delta_s = 1/(2(s+1))$ is the weight assigned to shifts (in either direction) of extent s, and $S(l)$ determines the shift range at position l.[2] In our applications, $S(l)$ is at most 20, hence the computational complexity of the kernel with shifts is only higher by a factor of at most 25.

From a mathematical point of view, it is important to ask the question whether this kernel is positive definite. If not, then the underlying SVM theory may not be applicable and optimization algorithms may fail. Let us suppose T is a shift operator, and Φ is the map associated with the zero-shift kernel k. Then the kernel $\tilde{k}(\mathbf{s}, \mathbf{s}') := \langle \Phi(\mathbf{s}) + \Phi(T\mathbf{s}), \Phi(\mathbf{s}') + \Phi(T\mathbf{s}') \rangle$ is trivially positive definite. However, we have $\tilde{k}(\mathbf{s}, \mathbf{s}') = \langle \Phi(\mathbf{s}), \Phi(\mathbf{s}') \rangle + \langle \Phi(T\mathbf{s}), \Phi(T\mathbf{s}') \rangle + \langle \Phi(T\mathbf{s}), \Phi(\mathbf{s}') \rangle + \langle \Phi(\mathbf{s}), \Phi(T\mathbf{s}') \rangle = k(\mathbf{s}, \mathbf{s}') + k(T\mathbf{s}, T\mathbf{s}') + k(T\mathbf{s}, \mathbf{s}') + k(\mathbf{s}, T\mathbf{s}')$. Assuming that we may discard edge effects, $k(T\mathbf{s}, T\mathbf{s}')$ is identical to $k(\mathbf{s}, \mathbf{s}')$; we then know that $2\,k(\mathbf{s}, \mathbf{s}') + k(T\mathbf{s}, \mathbf{s}') + k(\mathbf{s}, T\mathbf{s}')$ is positive definite. Our kernel (2), however, is a linear combination, with positive coefficients, of kernel of this type, albeit multiplied with different constants δ_s. The above arguments show that if δ_0 is at least twice as large as the sum of the remaining δ_s, the kernel will be positive definite. In our experiments detailed below, δ_0 does not in all cases satisfy this condition. Nevertheless, we have always found the kernel to be positive definite on the given training data, i.e. leading to positive definite matrices, and thus posing no difficulties for the SVM optimizer.

Finally, note that Meinicke *et al*. (2004) proposed a so-called 'oligo kernel' which is related to our extended WD kernel: for each possible $k$mer (k fixed in advance), one scans the sequence and generates a numeric sequence whose entries characterize the degree of match between the given kmer and the sequence (at the corresponding location). To achieve a certain amount of positional invariance, the numeric sequence is convolved with a Gaussian. The convolved sequences are concatenated to give the feature space representation of the original sequence. Meinicke *et al*. (2004) write down the dot product between two such feature space vectors, and observed that its evaluation does not require summing over all possible $k$mers occurring in the two sequences, but only over those that actually appear in both sequences. By construction as a dot product in some feature space, their kernel is positive definite. However, the kernel can be computationally demanding when $k$mers appear often in the sequences (e.g. for short $k$mers). Moreover, it only considers $k$mers of a fixed length rather than a mixture of $k$mers, which in our experience is a disadvantage for large k (which in turn is necessary in our application).

2.3 Distinguishing alternatively from constitutively spliced exons

2.3.1 Overview Alternatively, spliced exons have features that distinguish them from constitutively spliced exons. For instance, in Dror *et al*. (2004) features like the exon length, its divisibility by 3, the length of the flanking introns and the intensity of the polypyrimidine tract were used. Moreover, conservation patterns to another organism have been used, which has been one of the most discriminative features (Sorek and Ast, 2003). However, exons are not frequently conserved and the conservational features are not available. Here we aim to design a classifier that accurately distinguishes constitutive from alternatively spliced exons and only uses information that is always available and might also be used by the cellular splicing machinery, i.e. features derived from the exon and intron lengths and features based on the pre-mRNA sequence.

We propose to use the WD kernel as described in Section 2.2 to learn about sequence features near the 3′ and 5′ ends of the exon. We define a 201 nt window of $(-100, +100)$ around the acceptor and donor splice sites, respectively, and extract pairs of subsequences $(\boldsymbol{s}_{1,i}, \boldsymbol{s}_{2,i})$ for each exon $\mathbf{e}_i$, $i = 1, \ldots, N$. Each subsequence is used with its WD kernel for computing the similarity between examples (i.e. exons). In this way, the combined kernel captures positional information relative to the start and the end of the exon (particularly in the intronic regions upstream and downstream and the exonic sequence near the boundaries of the exon). The two WD kernels are linearly combined with a linear kernel on features $\mathbf{f}_i$ extracted

[2] Note that we could have already used γ_l in the position-dependent WD kernel described before.

from the exon and intron lengths:

$$k(\mathbf{e}_i, \mathbf{e}_j) = k_1(\boldsymbol{s}_{1,i}, \boldsymbol{s}_{1,j}) + k_1(\boldsymbol{s}_{2,i}, \boldsymbol{s}_{2,j}) + \sigma \langle \mathbf{f}_i, \mathbf{f}_j \rangle,$$

where σ is a scaling factor and $\mathbf{f}_i$ is a feature vector consisting of subvectors $\mathbf{f}_i^{el}, \mathbf{f}_i^{ilu}, \mathbf{f}_i^{ild}$ and $\mathbf{f}_i^{stp}$ characterizing the exon length $l(\mathbf{e}_i)$, upstream intron length $l(\mathbf{i}_i^u)$, downstream intron length $l(\mathbf{i}_i^d)$, and in which of the three frames of the exon stop codons appear, respectively. We define

$$(\mathbf{f}_i^{el})_j := \begin{cases} 1, & l(\mathbf{e}_i) \le L_j, \\ \frac{l(\mathbf{e}_i)-L_j}{L_{j+1}-L_j}, & L_j < l(\mathbf{e}_i) \le L_{j+1}, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

where the L_js are logarithmically spaced between 20 and 1000 nt (we use 30 bins).[3] We proceed analogously for the upstream and downstream intron lengths and let $\mathbf{f}_i^{stp}$ be a 3D vector with each dimension indicating whether the corresponding frame contains at least one stop codon. By defining the features in this way, the SVM learns a piecewise linear function of the lengths of the exons and introns. For k_1 and k_2, we use the extended WD kernel as defined in Equation (2) with a uniform weighting over the length of the sequence, i.e. $\gamma_l = 1/L(\boldsymbol{s})$, and the $\boldsymbol{\beta}$s and $\boldsymbol{\delta}$s as defined before. In addition, we use $S(l) = \kappa|p - l|$, where p is the position of the splice site: the farther away a motif is from the splice site, the less precisely it needs to be localized in order to contribute to the kernel value. Finally, we train the SVM using the kernel on 'positive exons', i.e. exons that show alternative splicing and 'negative exons' (i.e. exons that are constitutively spliced).

2.3.2 Model selection To estimate the out-of-sample-accuracy, we used a 5-fold cross-validation: the data were split into five disjoint subsets. Each set was used once to estimate the test error while the remaining four subsets were used for model selection and training. Model selection in turn is performed with respect to the true positive rate at a false positive rate[4] of 1% by a 5-fold cross-validation on this smaller set. We tune the SVMs regularization parameter C selecting $C \in \{0.5, 1, 2, 3, 5, 7, 10, 15, 20\}$, the WD kernels parameters $\kappa \in \{0, 0.05, 0.07, 0.1, 0.14, 0.19, 0.26, 0.37, 0.51, 0.72, 1\}$, $d \in \{5, 10, 15, 20, 23, 27, 30, 33, 37\}$, respectively. The window position around the donor and acceptor site is chosen to be $(-100, +100)$.[5] The scaling parameter $\sigma \in \{0.1/L, 1/L, 10/L\}$ was determined in the same cross-validation procedure, where $L = 201$ is the length of the sequence. For the optimized model parameters we retrained the SVM on the full training data (i.e. the remaining four sets of the outer cross-validation).

2.3.3 MKL for interpretation In order to understand which of the positions near the 3′ and 5′ ends of the exons are important, we use a recently proposed algorithm for MKL (Sonnenburg *et al*., 2005; see also Bach *et al*., 2004). The idea is to decompose the WD kernel (2) into positional subkernels, each defined only on a certain window around a position. For this work, we consider learning the weights γ_l in the kernel (2) (the weights $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$ are kept constant as before). For each position l one defines the kernel $k_l(\boldsymbol{s}, \boldsymbol{s}')$ as follows:

$$\sum_{k=1}^{d} \beta_k \sum_{\substack{s=0 \\ s+l \le L}}^{S(l)} \delta_s \left[\mathbf{I}(\boldsymbol{u}_{k,l+s}(\mathbf{s}) = \boldsymbol{u}_{k,l}(\mathbf{s}')) + \mathbf{I}(\boldsymbol{u}_{k,l}(\mathbf{s}') = \boldsymbol{u}_{k,l+s}(\mathbf{s}')) \right].$$

Finally, by MKL one finds the optimal $\boldsymbol{\gamma}$ for the convex combination of the positional subkernels

$$k(\boldsymbol{s}, \boldsymbol{s}') = \sum_{l=1}^{L-d} \gamma_l \, k_l(\boldsymbol{s}, \boldsymbol{s}'),$$

where $\sum_l \gamma_l = 1$ and $\gamma_l \ge 0$ $(l = 1, \ldots, L)$ (for details see Sonnenburg *et al*., 2005). (For this experiment we chose $d = 30$, $C = 1$, $\sigma = 0.1$ and $\kappa = 0.5$.)

The result is an importance weighting over the positions of the sequence that indicates which of the positions carry most discriminative information (shown in Section 3.1.2). Since existing MKL algorithms are currently still computationally too expensive to deal with hundreds of subkernels, we reduce the number of subkernels by blocking the weights—a resolution of 10 nt for the importance weighting seems sufficient for our purpose.[6] Hence, for the kernels taking into account the 3′ and the 5′ ends of the exon (each considering 201 nt windows), we get each 21 kernels leading to 42 weights to be determined in total. The learned weights are shown in Figure 3 (Section 3).

2.4 Finding skipped exons within introns

The algorithm proposed in the previous section is able to distinguish between constitutive exons and alternatively spliced exons. It can be applied for instance to already EST confirmed or predicted exons. However, this means that we can apply the method only if the exon is already known. In turn, if we want to apply the method, for instance, to EST confirmed regions, the likelihood is high that the exon is skipped in the existing sequencing results.[7] Moreover, gene prediction programs

[3]Most exon lengths are within this range and the choice of the number of bins is not sensitive.

[4]The true positive rate is the number of positively labeled examples identified as positive by the algorithm, divided by the total number of positively labeled examples. The false positive rate is the number of negatively labeled examples identified as positive, divided by the total number of negatively labeled examples.

[5]We also tested slightly longer and shorter windows not leading to significantly different results.

[6]Moreover we introduced a smoothing regularizer on the $\boldsymbol{\gamma}$s. Details would go beyond the scope of the paper.

[7]Without further Information, one could naively assume that one would pick (on average) the same fraction of mRNAs with and without the skipped exon.

may miss such alternatively spliced exons. We therefore propose an algorithm that not only classifies whether a certain exon is alternatively spliced, but it also locates it accurately within an intron. We later apply the algorithm to scan over all EST confirmed introns for skipped exons.

2.4.1 Splice site detection In order to find exons we need an accurate splice site classifier, which we briefly describe in the following. We start with a *C.elegans* set of about 110 000 cleaned EST sequences (similar to Section 2.1; described in more detail in the Supplementary material). Each match of the EST sequence to the genomic sequence defines several boundaries between exons and introns defining positive examples for acceptor and donor splice sites (∼40 000 of each site). We generated negative examples by considering all occurrences of the AG or GT dimer within introns and exons confirmed by ESTs, leading to ∼820 000 acceptor and 750 000 donor decoy examples, respectively. We followed the same procedure for EST sequences used to generate the alternative splice dataset (including positive and negative examples) to generate true and decoy acceptor and donor examples. We removed them from the above dataset in order to have independent sets. Of the remaining examples we use 70% for training, 10% for validation and 20% for testing (we split such that the sets are generated from non-overlapping regions of the genome). We use the standard WD kernel and train a SVM on the training data. Model selection for a wide range of regularization constants C, degrees d and window positions around the splice sites is performed using the validation set (as in Rätsch and Sonnenburg, 2003). On the test set we achieve an area under the curve (AUC) (Metz, 1978) of 99.75% and 99.74% for acceptor and donor site recognition, respectively.

2.4.2 Discriminative combination We now propose an algorithm able to accurately predict unknown and skipped exons within introns. The goal is to learn a scoring function that can classify potential exons within confirmed introns into real exons (that are then alternatively spliced) and false exons. Given this function, we only need to generate possible exon start/end pairs within the intron and can classify them using the scoring function. This method is particularly powerful when scanning over already EST confirmed introns for exon skip events. Surprisingly, it turns out that the problem of predicting whether there is a skipped exon within an intron can be solved more accurately than the previous problem, as we can exploit the very accurate splice site detection algorithm (Section 2.4.1). A previously missed exon has a certain characteristic: it either exhibits weak splice signals or should contain regulatory signals detectable by the algorithm in Section 2.3. We thus need a way to combine both predictions into a scoring function classifying exons with high accuracy.

In that case, exon skipping events in already confirmed introns (i.e. without any identified exon) would be equally likely as for exons that have already been found.

We can use our previous dataset of alternatively spliced and constitutive exons for this task. For truly alternatively spliced exons $\mathbf{e}_i$ the scoring function $f(\mathbf{e}_i^+)$ $(i = 1, \ldots, N^+)$ should return a positive score, while for all other possible exons $\mathbf{e}_{i,j}^+$ $(j = 1, \ldots, N_i^+)$ within the same intron it should return a negative score. For introns bordering constitutively spliced exons (there are two introns for each constitutively spliced exon triple in our database) the function should predict a negative value for possible exons $\mathbf{e}_i^-$ $(i = 1, \ldots, N^-$, $j = 1, \ldots, N_i^-)$ within the intron. Hence, we solve the following optimization problem where we enforce a large margin between positive and negative predictions:

$$\begin{aligned} \min_{\mathbf{w}} \quad & C\mathbf{P}(\mathbf{w}) + \sum_{i=1}^{N^+} \xi_i^+ + \sum_{i=1}^{N^-} \xi_I^-, \\ \text{s.t.} \quad & f_{\mathbf{w}}(\mathbf{e}_i^+) \geq 1 - \xi_i^+ \quad i = 1, \ldots, N^+, \\ & f_{\mathbf{w}}(\mathbf{e}_{i,j}^{\pm}) \leq -1 + \xi_I^- \quad i = 1, \ldots, N^-, \ j = 1, \ldots, N_i^-, \end{aligned}$$

where $f_{\mathbf{w}}$ is our scoring function parameterized by $\mathbf{w}$, the $\boldsymbol{\xi}$s are slack variables allowing for misclassifications and $\mathbf{P}(\mathbf{w})$ is a regularizer. We define the scoring function as $f(\mathbf{e}) = \langle \mathbf{w}, \mathbf{f}(\mathbf{e}) \rangle + w_0$. The feature vector $\mathbf{f}(\mathbf{e})$ for an exon $\mathbf{e}$ consists of similar components as in Section 2.3: characteristics of the lengths of the exon and the flanking introns and the occurrence of stop codons in the exons. Furthermore, it contains three components considering the scores of the SVM acceptor and donor splice site predictor (Section 2.4.1) and the recognizer for alternatively spliced exons (Section 2.3). We quantize the length and the SVM outputs similar to Equation (3), but to increase sparsity we used a slightly different feature vector[8]:

$$(\mathbf{f}(v))_j := \begin{cases} \frac{v - L_j}{L_{j+1} - L_j}, & L_j \leq v \leq L_{j+1}, \\ \frac{L_j - v}{L_j - L_{j-1}}, & L_{j-1} < v \leq L_j, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

We defined the regularizer $\mathbf{P}$ such that the piecewise linear functions with respect to the splice site scores and the alternative exon recognizer are monotonically increasing. Moreover, we penalized the piece-wise linear functions for exons and flanking intron lengths such that they have a small variation (sum of absolute differences from one step to the next). The resulting optimization problem has more than a million constraints and we used a column generation technique (Bennett *et al*., 2000) to efficiently solve the problem (∼2 h on a standard PC) with CPLEX (1994).[9]

[8]For the SVM outputs we used uniformly spaced support points between −3 and +4 in 30 bins.

[9]A few more details: (1) When generating exon candidates we only considered potential acceptor and donor sites where the SVM predicts at least a score of −3 (empirically determined to classify almost all positive examples as positive). (2) Since we used the same training set to train the alternative exon classification algorithm, the outputs, in particular, of the exons that are SVs are close to −1 or +1. We, therefore, used a leave-one-out scheme to estimate the output of the example at hand that would have been obtained when training without it (cf. Jaakkola and Haussler, 1999).

We trained our method on 75% of the available data in our alternatively spliced exon database. Evaluation is performed on the remaining 25% of the data. We only tested three values of C all leading to very similar results. We therefore did not perform cross-validation for model selection.

2.5 Material and methods for the biological confirmation experiment

We applied the alternative exon prediction algorithm described in Section 2.3 to a set of 21 508 exon triples that were confirmed exactly once by an EST (full coverage of the internal exon, flanking exons might only be partially covered). We use the proposed algorithm to predict whether the internal exon exhibits alternative splicing. For the top ranked 1% of the exons we tried to design primer pairs in the flanking exons (using Primer 3.0, Rozen and Skaletsky, 2000) such that the expected product size without exon is at least 150 nt long and the product with exon is not more than 500 nt in length. Since exons <50 nt would lead to a too small product size difference,[10] we had to exclude them from further analysis.[11] Out of the 215 sequences in the top 1% we were able to successfully design primers for 143 exons. We randomly chose 18 exons for further testing. We additionally included a negative control for an exon that is 10 times EST confirmed without evidence of alternative splicing and three positive controls for which each variant (i.e. with and without exon) is at least twice EST confirmed. A summary of the primers is listed on the Supplementary website.

We start with 20 μl randomly PCR amplified cDNAs (N^6 primers) provided by Waltraud Röseler (Max Planck Institute for Developmental Biology, Tübingen). For the verification of alternative splicing events, a typical PCR reaction mixture consisted of 10 mM Tris–HCl, 50 mM KCl, 1.5 mM $MgCl_2$, 200 μM dNTP, 1 M Betain, 1 unit *Taq* polymerase and 2 pmo/μl primer. PCR reaction and thermocycling was done in a Perkin Elmer Gene Amp 9700 PCR machine under standard conditions (40 cycles, 2″: 94°, 30″: 92°, 30″: 55°, 60″: 60°). The PCR products were first confirmed on a 1.5% agarose gel for their expected sizes. We did not proceed to the sequencing step, if only the larger product was confirmed on the gel (since we already have an EST for the case of exon inclusion). Once the length of at least two products was confirmed, they were extracted using a Qiagen Gel Extraction Kit. Sequencing reactions were set up according to the manufacturer's instructions for the Big Dye Terminator chemistry (Applied Biosystems, Foster City, CA). Samples were analyzed using capillary electrophoresis (Applied Biosystems, ABI Prism 3730). The software PHRED performed base calling and vector sequences were masked with CrossMatch. Sequences containing at least 100 non-vector bases with Phred values ≥20 were used for further analysis. The sequences obtained were then validated by aligning them using blast against the *C.elegans* genome (http://www.wormbase.org).

Once the gene identity was confirmed, we compared the gene structures of the obtained EST with our prediction. Only when there existed at least two products, one exhibiting splicing with exon and another one without the internal exon, we count the case as alternatively spliced. If only one product was found or the smaller product included the exon, we counted the case as false prediction. If we were unable to obtain a sufficiently long sequencing result for the smallest PCR product or if the PCR failed, then we excluded the case from the further evaluation.

[10] We required at least 20% difference in the size of the products (with and without internal exon).

[11] Note that we exclude quite a few good alternatively spliced exon candidates this way, since skipped exons are often short.

3 RESULTS AND DISCUSSION

3.1 Recognition of alternatively spliced exons

3.1.1 Simulation experiment The experiment was set up as described in Section 2.3.1. The parameters of the used SVM classifier were determined by cross-validation (Section 2.3.2). The tuned model parameters vary over the different test splits, e.g. $C \in \{0.5, 1, 2\}, \sigma \in \{1/L, 0.1/L\}$ and $d \in \{10, 15, 33\}$. By cross-validation we estimate the performance on unseen examples and achieve an AUC score of 89.74% and a true positive rate of 48.5% at a false positive rate of 1% (averaged over the five test splits). In Figure 2 the performance of our exon skip recognizer in terms of the (ROC) receiver operating characteristic curve (Metz, 1978) is displayed. The performance of our method compares well with the one reported in Dror *et al*., 2004 (true positive rate ∼ 50% at 0.5% false positive rate), given that we can apply it to arbitrary exons (and not just to the 25% conserved exons).

Our predictions for all single EST confirmed exons triples are found on the Supplementary website.

3.1.2 Understanding the SVM classifier To interpret the SVM classifiers result we employ MKL to determine the weights $\boldsymbol{\gamma}$ for each positional subkernel (Section 2.3.3) for both kernels used around the acceptor and donor sites. In Figure 3 the learned weighting is shown. A higher weight at a certain position in the sequence correspond to an increased importance of substrings starting at this location. Given this weighting, we can identify five regions which seem particularly important for discrimination: (a) and (b) within the upstream intron the region −70 to −40 and −30 to 0 (relative to the end of the intron), (c) the exon positions +30 to +70 (relative to the beginning of the exon) and (d) −90 to −30 (relative to the end of the exon). And finally, (e) the downstream intron positions 0–70 (relative to the beginning of the intron).

For these regions we counted the occurrence of all hexamers in the positive and negative examples. Using the frequency p of occurrence of a hexamer in the negative examples

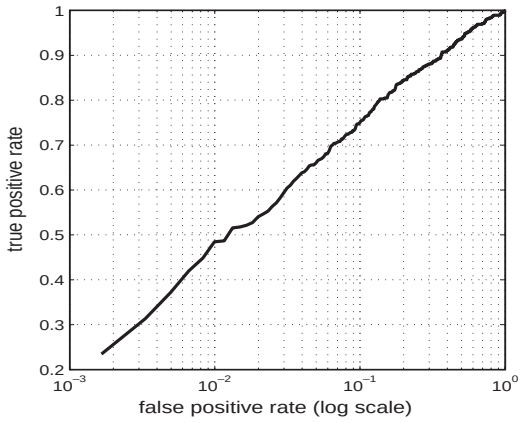

Fig. 2. ROC curve on the test set for the model found by cross-validation. The AUC is 89.74% and the true positive rate is 48.5% at a false positive rate of 1%. Note the logarithmic scale on the abscissa. All quantities are averaged over the five test splits.

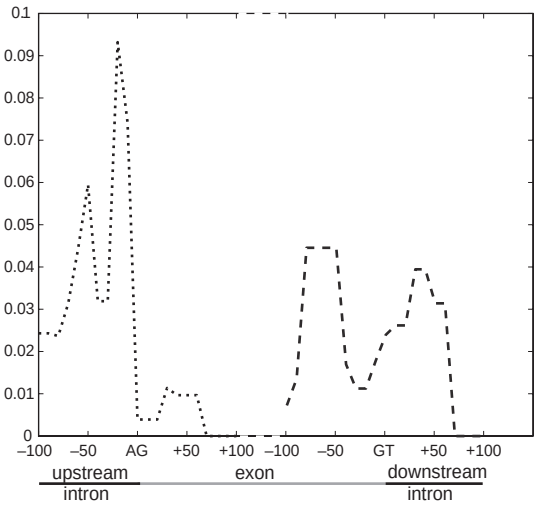

Fig. 3. We use MKL to determine weights for the WD kernel. Shown is learned weighting for the WD kernel at the acceptor and the donor sites. From areas of higher weight (upstream intron: regions −70 to −40 and −30 to 0, exon: +30 to +70 and −30 to −90, downstream intron 0 to +70), overrepresented hexamers have been extracted and are shown in Table 1.

as background model, we computed the *E*-value using the binomial distribution. In Table 1 we display for each of the five regions the six hexamers with highest *E*-value. Particularly interesting seem the motifs AGTGAG and CAGCAG which appear significantly only in the region near the exon start and exon end, respectively.

3.1.3 Biological validation We considered 21 508 exon triples (only single EST confirmed) for alternative splicing. For 18 randomly selected cases from the 1% top ranked predictions, we performed a confirmation experiment (Section 2.5).

Table 1. Shown are the top six ranked hexamers (by *E*-value) extracted for the upstream intron, the in-between exon and the following downstream exon

Upstream intron hexamer	*E*-value	Exon hexamer	*E*-value	Downstream intron hexamer	*E*-value
(a) Relative to exon start					
CTAACC	1.2e−17	**AGTGAG**	4.2e−11	TGTGTG	5.9e−31
CCCCCC	3.8e−11	TTTTTT	2.7e−9	TTGTGT	1.7e−24
TAACCC	9.8e−10	ATATAT	1.3e−8	GTGTGT	3.6e−16
CACTTT	6.2e−9	TATATA	3.6e−7	GTTGTG	4.4e−15
ATCCCC	1.6e−0	ATAGGT	4.8e−7	TGTTGT	3.3e−14
CTTTCC	2.4e−7	TAGGTT	5.0e−7	TGCATG	1.3e−13
(b) Relative to exon end					
CATTCT	1.3e−9	TTTAAA	1.8e−12		
CTCTCT	1.9e−9	AATTTT	2.2e−10		
GCATGT	4.4e−9	ATTTTA	2.9e−9		
GTTGTC	4.4e−9	**CAGCAG**	1.2e−8		
TCTCTA	2.2e−8	TAATTT	8.3e−8		
CTCTAT	1.1e−7	TTCCCC	2.1e−7		

The first column in part (a) shows the most important hexamers in the intron for the region −70 to −40 relative to the end of the intron. Part (b) states hexamers contained −30 bp until the end of the upstream intron. Similarly, the second column displays hexamers in the exon from +30 to +70 (a) and −30 to −90 (b) and the last column hexamers in the downstream intron from 0 to +70.

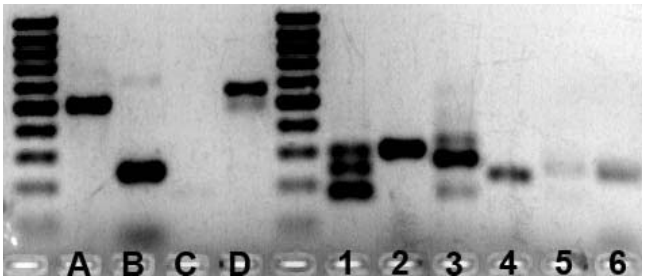

Fig. 4. The gel electrophoresis plot obtained in the wetlab experiment. Only the control sequences A, B, C, D (B–D are positive controls) and the first six evaluated sequences are shown. Images of the gels for the remaining sequences 7–16 are found on the Supplementary web page.

In 11 experiments we obtained at least two PCR products of appropriate size, while in other 5 cases we obtained only one PCR product (Fig. 4). In remaining two cases the PCR failed and did not lead to a measurable product. For the negative control, we correctly obtained only one product and for two of the three positive controls, we obtained two products (PCR failed for the third). For 11 test cases and the 2 positive controls we sequenced the different PCR products and obtained 6 significant sequencing results (including 1 for a positive control).[12] Out of the five significant test cases three exhibited alternative exon usage (verified by aligning the sequences against the genome). Unfortunately, the sequenced products

[12]For more details see the Supplementary website.

for the remaining positive control did not show evidence for alternative splicing although the exon is known to be alternatively spliced. This indicates that the biological testing set up is not yet optimal, and that further scrutiny might well reveal that more of the candidates predicted by our algorithm do indeed show alternative splicing. More experimental data are available on the Supplementary website.

Hence, out of 10 significant results (5 with 1 PCR product and 5 cases with at least 2 sequenced PCR products) we found 3 truly alternatively spliced exons (30%). If one assumes that 1% of all exons are alternatively spliced and our algorithm achieves a true positive rate of 50% at a false positive rate of 1%, then one would expect that out of three predicted exons one is true positive and two are false positive exons, and hence only 33% of those exons would be true positive. This Gedanken experiment supports the experimentally observed performance. Furthermore, note that if only 0.5% of all exons are alternatively spliced, we would in the best case only, find alternatively spliced exons in the top 0.5% of our ranking. In fact, the three correctly predicted alternatively spliced exons are among the first five significant exons. This indicates that the expected fraction of true alternative exons in the top 0.5% is considerable higher (60% in our experiment).

3.2 Finding skipped exons within introns

We trained the algorithm proposed in Section 2.4 on 4161 introns of which 365 contained an alternatively spliced exon (regularization constant $C = 1$). We evaluated the algorithm on the remaining 1388 introns (including 122 introns containing alternatively spliced exons), which were not used for training. In the first level of evaluation, we let the algorithm predict whether there is a skipped exon in the intron. The outcome should be positive for introns containing an alternatively spliced exon and negative for constitutively spliced introns (i.e. not containing any skipped exon). For this task, the algorithm achieves an AUC of 99.0 and identifies 85.2% of all introns that contain alternatively spliced exons at a false positive rate of 1% (Fig. 5). Note, however, that out of the 1388 test introns ∼62% were shorter than 60 nt such that no exon could fit into the intron (introns are always observed to be longer than 30 nt). If we restrict our attention only to those introns that contain at least one possible exon candidate (with not too weak acceptor and donor splice sites), then the AUC score drops slightly to 97.2%.

In the second evaluation, we are not only interested whether the intron contains an alternatively spliced exon but also whether our algorithm predicted the correct exon. Out of the 122 introns containing an alternatively spliced exon, the algorithm identified the true exon in 90 cases (73.8%) as the one with the maximum score (among all possible exons within the intron; not necessarily above some threshold). In the remaining cases, the algorithm predicts another exon with a larger score, which in fact could be another alternatively spliced exon that has not yet been found (cascade exons). For the native threshold w_0 obtained by the learning algorithm, we accurately classified 70 alternatively spliced exons (57.4%) above the threshold (not necessarily the one with maximal score).[13]

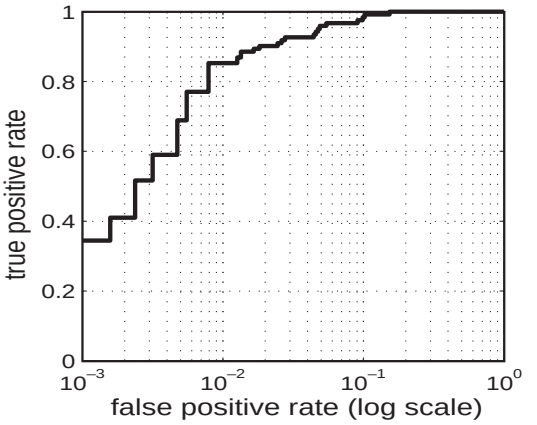

Fig. 5. ROC curve obtained for the classifier that detects skipped exons within introns. We achieve an AUC of 99.0%. At a level of 1% false positives, we detect 85.2% true positives and at 0.5% false positives we recognize 68.9% of the alternatively spliced exons.

These experiments show that it is actually easier to find alternatively exons that were skipped, for instance, in existing sequencing databases. Our algorithm not only accurately predicts whether an intron contains a yet unknown alternatively spliced exon but also locates it with rather high accuracy within the intron. Note that our method is also applicable to find, for instance, exclusively used exons.

Our predictions for all single EST confirmed introns are found on the supplementary website.

4 CONCLUSION

We have presented a system for identifying alternatively spliced exons, and confirmed its validity by computational experiments and (so far limited) wetlab experiments. The system is applicable independent of the availability of phylogenetic information. From a biological point of view, this is attractive, since our system uses only information which is in principle available to the cellular splicing mechanism. From a computational point of view, it has the advantage that it can also be applied in cases where no phylogenetic information exists. Although it makes sense to use such information if it is available (and we are planning to include it into our system), this feature of our method makes it applicable in a larger domain.

We applied our system to all available exonic and intronic regions of *C.elegans* that were ESTs confirmed and identified a large number of potential alternatively spliced exons (the list

[13]Note that this threshold can easily be lowered leading to a larger fraction of accurately identified alternatively spliced exons.

is available on the website). Combined with large scale biochemical verification experiments these predictions are likely to help to uncover the full transcriptome of *C.elegans*.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge partial support from the PASCAL Network of Excellence (EU #506778), DFG grants JA 379/13-2 and MU 987/2-1. We thank Alexander Zien, Koji Tsuda, K.-R. Müller, Ralf Sommer, Lars Knoch, Dirk Holste and Gene Yeo for the helpful and motivating discussions. Many thanks to Waltraud Röseler and Ralf Sommer at the Department of Developmental Biology in Tübingen for generously providing *C.elegans* cDNAs. We are grateful to the referees who helped improving the manuscript in many ways. Finally, thanks to Thomas Jeffke at AGOWA Berlin for his great efforts and the good collaboration.

REFERENCES

Bach,F.R., Lanckriet,G.R.G. and Jordan,M.I. (2004) Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the 21st International Conference on Machine Learning*, (*ICML'04*), ACM Press.

Bennett,K.P., Demiriz,A. and Shawe-Taylor,J. (2000) A column generation algorithm for boosting. In *Proceedings of the International Conference on Machine Learning*, (*ICML'00*), San Francisco, Morgan Kaufmann, pp. 65–72.

Boguski,M.S., Lowe,T.M. and Tolstoshev,C.M. (1993) dbEST—database for ESTs. *Nat. Genet.*, **4**, 332–333.

Burge,C. and Karlin,S. (1997) Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.*, **268**, 78–94.

CPLEX Optimization Incorporated (1994) *Using the CPLEX Callable Library*. Incline Village, Nevada.

Dror,G., Sorek,R. and Shamir,R. (2004) Accurate identification of alternatively spliced exons using support vector machine. *Bioinformatics*, **21**, 897–901.

Gupta,S., Zink,D., Korn,B., Vingron,M. and Haas,S. (2004) Strengths and weaknesses of EST-based prediction of tissue-specific alternative splicing. *BMC Genomics*, **5**, 72.

Harris,T.W., Chen,N., Cunningham,F., Tello-Ruiz,M., Antuschechkin,I., Bastiani,C., Bjeri,T., Blasiar,D., Bradnum,K., Chan,J. *et al*. (2004) Wormbase: a multi-species resource for nematode biology and genomics. *Nucleic Acids Res.*, **32**, D411–D417.

Haussler,D. (1999) Convolutional kernels on discrete structures. *Technical Report CRL-99-10*, UC Santa Cruz.

Hiller,M., Backofen,R., Heymann,S., Busch,A., Glaesser,T.M. and Freytag,J.C. (2004) Efficient prediction of alternative splice forms using protein domain homology. *In Silico Biol.*, **4**, 195–208.

Jaakkola,T.S. and Haussler,D. (1999) Probabilistic kernel regression models. In *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics, San francisco, CA. AISTAT'99*.

Kent,W.J. (2002) Blat—the blast-like alignment tool. *Genome Res.*, **12**, 656–664.

Leslie,C., Eskin,E. and Noble,W.S. (2002) The spectrum kernel: a string kernel for SVM protein classification. In *Proceedings of the 7th Pacific Symposium of Biocomputing PSB'02*, Kaua'i, Hawaii, 2002.

Maniatis,T. and Tasic,B. (2002) Alternative pre-mRNA splicing and proteome expansion in metazoans. *Nature*, **418**, 236–243.

Meinicke,P., Tech,M., Morgenstern,B. and Merkl,R. (2004) Oligo kernels for datamining on biological sequences: a case study on prokaryotic translation initiation sites. *BMC Bioinformatics*, **5**, 169.

Metz,C.E. (1978) Basic principles of ROC analysis. *Seminars Nucl. Med.*, **8**(4), 283–298.

Rätsch,G. and Sonnenburg,S. (2003) Accurate splice site prediction for *Caenorhabditis elegans*. In Schölkopf,B., Tsuda,K. and Vert,J.P. (eds.), *Kernel Methods in Computational Biology*, MIT Press, Cambridge, MA, pp. 277–298.

Rozen,S. and Skaletsky,H.J. (2000) Primer3 on the WWW for general users and for biologist programmers. In Krawet,S. and Misener,S. (eds), *Bioinformatics Methods and Protocols: Methods in Molecular Biology*. Humana Press, Totowa, NJ, pp. 365–386.

Sakai,H. and Maruyama,O. (2004) Extensive search for discriminative features of alternative splicing. In *Proceedings of the Pacific Symposium on Biocomputing*, Hawaii, USA, pp. 54–65.

Schölkopf,B., Tsuda,K. and Vert,J.P. (eds) (2004). *Kernel Methods in Computational Biology*, MIT Press series on Computational Molecular Biology. MIT Press, Cambridge, MA.

Sonnenburg,S., Rätsch,G. and Schäfer,C. (2005) Learning interpretable SVMs for biological sequence classification. In *Proceedings of the 9th International Conference on Research in Computational Molecular Biology (RECOMB 2005)*, LNBI 3500, Springer Berlin, pp. 389–407.

Sorek,R. and Ast,G. (2003) Intronic sequences flanking alternatively spliced exons are conserved between human and mouse. *Genome Res.*, **13**, 1631–1637.

Sorek,R., Shemesh,R., Cohen,Y., Basechess,O., Ast,G. and Shamir,R. (2004) A non-EST-based method for exon-skipping prediction. *Genome Res.*, **14**, 1617–1623.

Vapnik,V.N. (1995) *The Nature of Statistical Learning Theory*. Springer Verlag, New York.

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

Wheeler,D.L., Church,D.M., Federhen,S., Lash,A.E., Madden,T.L., Pontius,J.U., Schuler,G.D., Schriml,L.M., Sequeira,E., Talusova,T.A. and Wagner,L. (2003) Database resources of the national center for biotechnology. *Nucleic Acids Res.*, **31**, 28–33.