BIOINFORMATICS ORIGINAL PAPER
Vol. 21 no. 8 2005, pages 1332–1338
doi:10.1093/bioinformatics/bti166

Genome analysis

SpliceMachine: predicting splice sites from high-dimensional local context representations

Sven Degroeve[1,*], Yvan Saeys[1], Bernard De Baets[2], Pierre Rouzé[3] and Yves Van de Peer[1]

[1]Department of Plant Systems Biology, Flanders Interuniversity Institute for Biotechnology (VIB), Technologiepark 927, Gent 9052, Belgium, [2]Department of Applied Mathematics, Biometrics and Process Control, Ghent University, Coupure links 653, Gent 9000, Belgium and [3]Laboratoire associé de l'INRA (France), Technologiepark 927, Gent 9052, Belgium

Received on August 23, 2004; revised on October 25, 2004; accepted on November 18, 2004
Advance Access publication November 25, 2004

ABSTRACT

Motivation: In this age of complete genome sequencing, finding the location and structure of genes is crucial for further molecular research. The accurate prediction of intron boundaries largely facilitates the correct prediction of gene structure in nuclear genomes. Many tools for localizing these boundaries on DNA sequences have been developed and are available to researchers through the internet. Nevertheless, these tools still make many false positive predictions.

Results: This manuscript presents a novel publicly available splice site prediction tool named SpliceMachine that (i) shows state-of-the-art prediction performance on *Arabidopsis thaliana* and human sequences, (ii) performs a computationally fast annotation and (iii) can be trained by the user on its own data.

Availability: Results, figures and software are available at http://www.bioinformatics.psb.ugent.be/supplementary_data/

Contact: sven.degroeve@psb.ugent.be; yves.vandepeer@psb.ugent.be

INTRODUCTION

An increasingly important task in bioinformatics is to analyze genome sequences for the location and structure of their genes, often referred to as gene prediction or gene finding. For most eukaryotic nuclear genomes, a gene usually consists of a set of coding fragments, known as exons, which are separated by non-coding intervening fragments, known as introns. The boundaries of these introns are called the splice sites, the 5′ boundary is termed the donor site and the 3′ boundary is termed the acceptor site.

Current gene prediction systems tend to have a modular structure, combining the outputs of several components that are each specialized in recognizing specific structural elements of a gene (Mathé *et al*., 2002). An important component is the splice site predictor. Computationally speaking, predicting the location of a splice site can be seen as a classification task. Although many eukaryotic organisms contain two kinds of spliceosomes splicing two types of introns, U2-type and U12-type, the vast majority of introns are U2-type (Patel *et al*., 2003) where the donor site practically always contains the GT dinucleotide at the intron boundary, GC being observed in less than 1% of the cases. This donor site is recognized by the U1 snRNA of the spliceosome through base-pairing with an ACUUACCU motif, and should ideally have the AG/GTAAGT pattern. Nevertheless, the base-pairing recognition is rather loose, i.e. the donor site pattern is less clear and tolerates many replacements in the motif, except for the border GT. The acceptor is observed to always contain the AG dinucleotide at the intron border with an even less clear pattern surrounding the dinucleotide. As such, all GT (resp. AG) dinucleotides on the DNA are defined as candidate donor (resp. acceptor) sites and need to be classified as either an actual (true) site or a pseudo (false) site.

Through the fast pace of the sequencing of genes and their cognate transcripts, the number of experimentally identified eukaryotic donor and acceptor sites has grown extensively over the last decade. The accumulation of publicly available biological data has boosted genomic research in the field of Machine Learning and the prediction of splice sites became again a challenge (Cai *et al*., 2000; Dash *et al*., 2001; Yeo *et al*., 2003; Castelo *et al*., 2004). Recent approaches based on discriminant functions such as Winnow (Chuang *et al*., 2001) or the support vector machine (SVM) (Sonnenburg *et al*., 2002; Degroeve *et al*., 2002; Sun *et al*., 2003) show significant improvements in prediction performance compared to previously used systems such as NetGene2 (Tolstrup *et al*., 1997), SPL, SplicePredictor (Usuka *et al*., 2000) and GeneSplicer (Pertea *et al*., 2001). Nevertheless, these approaches have not yet been implemented as a tool that can be used by researchers for annotating genome sequences.

This manuscript presents a novel publicly available splice site prediction tool named SpliceMachine that (i) shows state-of-the-art prediction performance on *Arabidopsis thaliana* and human sequences, (ii) can be trained by the user on its own data, (iii) performs a computationally fast annotation and (iv) is intuitive and can provide biological knowledge extracted from the data. Our approach employs linear support vector machines (LSVM) to compute a linear classification boundary between actual and pseudo splice sites. For this, a candidate splice site is represented as a feature vector, each feature containing some information about the candidate splice site and its context in the sequence. This context is defined as the subsequence that starts at p nucleotide positions upstream of the candidate splice site and ends at q positions downstream of the candidate splice site.

*To whom correspondence should be addressed.

We define feature sets in order to capture the positional nucleotide preferences observed in close proximity to the donor and acceptor site, the preference for certain oligomers in the neighborhood of splice sites (Lim *et al*., 2001), and the codon bias upstream donor and downstream acceptor sites. We propose a model-based procedure for optimizing the parameters p and q for each of these feature sets as well as the cost parameter C of the LSVM (described further) and show that this leads to a significant boost in the prediction performance of the system.

METHODS AND DATA

Local context representations

This section describes the feature vector representations of candidate splice sites. The first type of features refers to *positional information* and should capture the consensus motif as well as the correlations that exists between nucleotide positions in close proximity of the splice site (see Introduction). The second type of features refers to *compositional information* and should capture the presence or absence of discriminative oligomers found in the neighborhood of splice sites. The existence of these oligomers was observed for instance by Lim *et al*. (2001) and used as discriminative information in SplicePredictor. A third type of information is the codon bias that exists upstream of most donor and downstream of most acceptor sites. This codon bias is also found downstream of pseudo donor sites or upstream of pseudo acceptor sites that are extracted from the coding part of a gene. This type of information is termed *coding potential* and it is similar to the compositional information but split up in each of the three possible reading frames as explained further.

All of the features are extracted from a local context subsequence surrounding the candidate splice site. This local context consists of p adjacent nucleotides upstream and q adjacent nucleotides downstream of the GT or AG dinucleotide. Throughout this paper, the following local context ($p = 6$, $q = 7$) around a candidate donor site is used to exemplify the terminology introduced:

```
a c t t c g G T a g c  c  t  c  c
1 2 3 4 5 6    7 8 9 10 11 12 13
```

Positional information The information extracted is the presence or absence of a nucleotide at a position in the local context subsequence. We will refer to this feature set as P1. Let $f_{s,v}$ be a binary feature from P1 that has value 1 if the nucleotide at position s in the local context sequence is v with $v \in \{a, c, g, t\}$. For the candidate donor site (see example) the following features in P1 have a value equal to 1 (all other features have value 0):

$$f_{1,\mathrm{a}}, f_{2,\mathrm{c}}, f_{3,\mathrm{t}}, f_{4,\mathrm{t}}, f_{5,\mathrm{c}}, f_{6,\mathrm{g}}, f_{7,\mathrm{a}}, f_{8,\mathrm{g}}, f_{9,\mathrm{c}}, f_{10,\mathrm{c}}, f_{11,\mathrm{t}}, f_{12,\mathrm{c}} \text{ and } f_{13,\mathrm{c}}.$$

To account for the correlations that exist between nucleotide positions, certain types of concatenations are created using the features in P1. Feature sets P2 and P3 extract the presence or absence of a di- or tri-nucleotide at a position in the local context subsequence. Let v be a di-nucleotide; then for the feature set P2 (similarly for the feature set P3) the following features have a value equal to 1:

$$f_{1,\mathrm{ac}}, f_{2,\mathrm{ct}}, f_{3,\mathrm{tt}}, f_{4,\mathrm{tc}}, f_{5,\mathrm{cg}}, f_{7,\mathrm{ag}}, f_{8,\mathrm{gc}}, f_{9,\mathrm{cc}}, f_{10,\mathrm{ct}}, f_{11,\mathrm{tc}}, f_{12,\mathrm{cc}}.$$

Compositional information The information extracted is the presence or absence of individual tri-, tetra-, penta- or hexamers in the local upstream context and in the local downstream context separately. We will refer to these feature sets as Ck where k is the length of the oligomer (k-mer) that is considered. Each feature set Ck consists of 4^k features $f_{\mathrm{up},x}$ that indicate the presence or absence of a k-mer x in the upstream context plus 4^k features $f_{\mathrm{down},x}$, that indicate the presence or absence of a k-mer x in the downstream context. For the candidate donor example the following compositional trimer features (C3) have a value equal to 1:

$$f_{\mathrm{up,act}}, f_{\mathrm{up,ctt}}, f_{\mathrm{up,ttc}}, f_{\mathrm{up,tcg}}, f_{\mathrm{down,agc}}, f_{\mathrm{down,gcc}}, f_{\mathrm{down,cct}}, f_{\mathrm{down,ctc}} \text{ and } f_{\mathrm{down,tcc}}.$$

Coding potential The information extracted is the presence or absence of codons in each of the three possible reading frames of both local upstream and downstream contexts separately. Let us assume that the complete context of our candidate site is a coding sequence, i.e. the candidate site would be a pseudo splice site. We can then write this context sequence in each of the three possible reading frames. For each of the three reading frames $R (R = 1, 2, 3)$ 64 features $f_{R,\mathrm{up},x}$ are computed for the upstream context plus 64 features $f_{R,\mathrm{down},x}$ for the downstream context. A feature $f_{R,\mathrm{up},x}$ has value 1 if the upstream context in reading frame R contains the trimer or codon x. This totals $128 \times 3 = 384$ features in a set we will denote RF (for Reading Frame). For the candidate donor site presented above the following features in RF have a value equal to 1:

$$f_{1,\mathrm{up,act}}, f_{1,\mathrm{up,tcg}}, f_{2,\mathrm{up,ctt}}, f_{3,\mathrm{up,ttc}}, f_{1,\mathrm{down,agc}}, f_{1,\mathrm{down,ctc}}, f_{2,\mathrm{down,gcc}}, f_{2,\mathrm{down,tcc}} \text{ and } f_{3,\mathrm{down,cct}}.$$

The definition of which reading frame is frame R is irrelevant as long as it is the same for all candidate sites.

Linear support vector machines

The SVM (Boser *et al*., 1992; Vapnik, 1995) is a data-driven method for solving two-class classification tasks. The LSVM separates the two classes in T with a hyperplane in the feature space such that:

(a) the largest possible fraction of instances of the same class are on the same side of the hyperplane, and

(b) the distance of either class from the hyperplane is maximal.

The prediction of an LSVM for an unseen instance $\mathbf{z}$ is 1 (classified as a positive instance) or -1 (classified as a negative instance), given by the decision function

$$\mathrm{pred}(\mathbf{z}) = t(\mathbf{w}\mathbf{z} + b) \tag{1}$$

with $t(x)$ a function that maps all x greater or equal to a certain threshold to 1 and all x smaller than that threshold to -1. The hyperplane is computed by maximizing a vector of Lagrange multipliers α in

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x_i} \mathbf{x_j},$$

$$\text{constrained to:} \quad 0 \le \alpha_i \le C \quad \text{and} \quad \sum_{i=1}^{l} \alpha_i y_i = 0, \tag{2}$$

where C is a parameter set by the user to regulate the effect of outliers and noise; i.e. it defines the meaning of the word largest in (a).

For the LSVM the relation between $\mathbf{w}$ and α is:

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i \mathbf{x_i} y_i.$$

Data

In a first benchmark, we compiled our own *A.thaliana* splice site data set and carefully checked whether no genes were included that were also in the test set called AraSet (Pavy *et al*., 1999). AraSet is a set of 168 *A.thaliana* genes which was used to compare several splice site prediction tools. Our training set was generated by aligning mRNAs [using SIM4; Florea *et al*. (1998)], obtained from the public EMBL database (June 5, 2000), with the BAC sequences that were used for the *Arabidopsis* chromosome assembly. Redundant genes were excluded by counting the neighbors of every gene (two genes are neighbors when they show more than 80% identity at the nucleotide level), and discarding the gene with the largest number of neighbors. This

process is repeated until no genes with neighbors remain. Of the 1812 genes obtained from EMBL (Aubourg *et al*., unpublished), 1495 genes were kept after removing redundant ones. From each gene only these introns confirming the GT–AG consensus were used to construct the set of actual splice sites. The pseudo donor sites were, for all genes, defined as all GT dinucleotides that are located between 300 nucleotide positions upstream of the translation start site and 300 nucleotide positions downstream of the translation stop site in that gene and that are not donor sites. The pseudo acceptor sites are defined as all AG dinucleotides within the same range and that are not acceptor sites. A sub-sample of this data set will be used for optimizing the parameters for *Arabidopsis*. The full data set was used to induce the *Arabidopsis* models in SpliceMachine. Training on this set and testing on AraSet will be refered to as benchmark B1.

In a second benchmark, we used the 1323 *Arabidopsis* genes and the 1115 human genes that were used to train and evaluate the GeneSplicer system in Pertea *et al*. (2001). A sub-sample of the human data set was used for optimizing the parameters for human gene annotation. The full human data set was used to induce the final human models in SpliceMachine. Training and testing on these sets will be referred to as benchmark B2ath for *Arabidopsis* and benchmark B2hum for humans.

Performance measures

Several measures have been used to evaluate prediction performance. In B1 the authors used sensitivity (Se) and specificity (Sp) rate defined as

$$\mathrm{Se} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \quad \text{and} \quad \mathrm{Sp} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}},$$

where TP, FP and FN are the number of true positives, false positives and false negatives, respectively. The performance of GeneSplicer in B2ath and B2hum was measured in terms of Se and false positive rate defined as

$$\mathrm{FP\%} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}.$$

By varying the decision threshold used to map Equation (1) onto a class, Sp and FP% ratios can be computed for all Se levels. For the model-based procedure used to optimize the parameters p, q and C (see further) the Sp ratio at 5% false negative predictions (Se = 0.95) is used as the criterion to measure prediction performance. This measure will be referred to as FN5%. For the train-test split of the data set the m-fold cross-validation procedure (mCV) is applied. In this setting the data is divided into m subsets of equal size while preserving the class distribution. A model is induced m times, each time leaving out one of the subsets from training that is then used to compute the performance measures as describe above.

RESULTS AND DISCUSSION

Parameter optimization

The optimization of the parameters that are associated with each of the representations is considered to be an important part in the computation of the SpliceMachine models. These parameters are the cost C [Equation (2)] used for training the LSVM, the length of the local context subsequence (p, q) for each of the feature sets as well as the optimal merging of feature sets in terms of classification performance.

For the parameter C we consider the values $2^{-12}, 2^{-11}, \ldots, 1, \ldots, 2^4$. The context lengths p and q can both take values in $\{20, 40, 60, 80, 100\}$. For each feature set the FN5% ratio is computed using 10 CV for all possible combinations (p, q, C). Although the LSVM induces a classifier relatively quick, a smaller data set needs to be randomly selected from the full data set to make the model-based optimization procedure practical. For both *Arabidopsis* and human we used a sub-sample that contains 1000 actual and 10 000 pseudo sites. For each feature set the (p, q) values for which FN5% is maximal (first row, in bold) and the (p, q) values for which FN5% is minimal (last row) are plotted in Table 1. To show that choosing large context sizes ($p = q = 100$) does not always lead to the best prediction performance, the results for $p = q = 100$ are plotted in the second row for each feature set. The third row shows performance for $p = q = 50$, a typical context size used in splice site prediction literature. In some cases the optimal value for p or q was at the border of the search space (=100). In these situations the search space for the context size that was at the border was increased by 20 positions until no further improvement was observed. For all feature sets the optimization of p and q shows a significant increase in 10CV prediction performance for most data sets.

Table 1 also shows that the influence of context size optimization is more pronounced in the case of the positional invariant feature sets C*k* and RF. For instance for B2hum acceptor site prediction, Table 1 shows that optimizing the context size increases 10CV prediction performance from 0.21 ($p = q = 50$) to 0.44 ($p = 20$, $q = 100$). These large differences in FN5% performance make sense because the value of a feature in these positional invariant feature sets depends strongly on p and q, while for P1, P2 and P3 this is not the case. For the compositional feature sets, C4 (words of length 4) shows the best overall performance. For acceptor site prediction using the feature sets C*k*, Table 1 shows optimal context sizes to be larger in the exon part of the local context subsequence. For donor prediction this is only the case for the human data sample. Although the differences are not necessarily species-dependent (other factors could be the sub-sampling of data points or the level of noise in the data) optimizing the context lengths for each genome (or new data set) is shown to be crucial for the induction of accurate species-specific prediction models. The default p and q values $p = q = 50$ seems to perform better than $p = q = 100$.

From Table 1 the optimal (p, q, C) combination for each of the feature sets can be obtained (the row in bold). If the feature sets represent different types of discriminative information, then the merging of feature sets (using more than one feature set to represent a candidate splice site by concatenating the features) should increase splice site prediction performance. But merging the feature sets will increase the information redundancy that could, in turn, decrease prediction performance (Kohavi *et al*., 1997). To investigate this, a larger data sub-sample was randomly extracted from the full data sets (B1 and B2) that contains 1250 actual and 50 000 pseudo sites. Feature sets are merged as follows:

merge$_{50-50}$: all feature sets P1, P2, P3, C3, C4, C5, C6 and RF using $p = q = 50$.

merge$_{50-50-C\mathit{best}}$: feature sets P1, P2, P3, C*best* and RF using $p = q = 50$. C*best* is the best performing C*k* from Table 1. This is C4 for all data sets.

merge$_{100-100}$: all feature sets P1, P2, P3, C3, C4, C5, C6 and RF using $p = q = 100$.

merge$_{100-100-C\mathit{best}}$: feature sets P1, P2, P3, C*best* and RF using $p = q = 100$. C*best* is the best performing C*k* from Table 1. This is C4 for all data sets.

merge$_{\mathrm{opt-opt}}$: all feature sets P1, P2, P3, C3, C4, C5, C6 and RF using the optimal values for p and q from Table 1.

merge$_{\mathrm{opt-opt-C}\mathit{best}}$: feature sets P1, P2, P3, C*best* and RF using the optimal values for p and q from Table 1. C*best* is the best performing C*k* from Table 1. This is C4 for all data sets.

Table 1. Optimal context sizes (p, q) for the different feature sets P1–P3 (positional), C3–C6 (compositional) and RF (coding potential) for both *Arabidopsis* and human sequences

	Arabidopsis (B1)						Humans (B2hum)					
	Donors			Acceptors			Donors			Acceptors		
	p	q	FN5%	p	q	FN5%	p	q	FN5%	p	q	FN5%
P1	**60**	**120**	**0.70**	**80**	**80**	**0.57**	**20**	**20**	**0.52**	**60**	**60**	**0.44**
	100	100	0.66	100	100	0.55	100	100	0.46	100	100	0.38
	50	50	0.68	50	50	0.49	50	50	0.49	50	50	0.44
	20	20	0.54	20	100	0.42	20	100	0.45	100	100	0.39
P2	**40**	**120**	**0.67**	**80**	**60**	**0.52**	**20**	**20**	**0.56**	**20**	**20**	**0.44**
	100	100	0.61	100	100	0.42	100	100	0.43	100	100	0.35
	50	50	0.65	50	50	0.47	50	50	0.49	50	50	0.44
	100	20	0.49	20	80	0.41	20	100	0.42	100	20	0.34
P3	**40**	**80**	**0.51**	**80**	**60**	**0.39**	**20**	**20**	**0.47**	**20**	**20**	**0.37**
	100	100	0.47	100	100	0.32	100	100	0.36	100	100	0.35
	50	50	0.48	50	50	0.38	50	50	0.44	50	50	0.34
	80	40	0.43	20	100	0.26	20	80	0.36	80	20	0.29
C3	**20**	**80**	**0.19**	**20**	**60**	**0.33**	**80**	**20**	**0.16**	**20**	**100**	**0.40**
	100	100	0.15	100	100	0.16	100	100	0.12	100	100	0.12
	50	50	0.18	50	50	0.24	50	50	0.13	50	50	0.19
	100	20	0.15	100	20	0.14	20	100	0.11	100	20	0.11
C4	**80**	**80**	**0.24**	**20**	**80**	**0.34**	**80**	**20**	**0.21**	**20**	**100**	**0.44**
	100	100	0.21	100	100	0.22	100	100	0.16	100	100	0.18
	50	50	0.24	50	50	0.28	50	50	0.17	50	50	0.24
	60	20	0.17	100	40	0.22	20	80	0.14	100	20	0.13
C5	**80**	**80**	**0.24**	**40**	**60**	**0.29**	**80**	**20**	**0.19**	**20**	**80**	**0.42**
	100	100	0.21	100	100	0.22	100	100	0.17	100	100	0.20
	50	50	0.21	50	50	0.26	50	50	0.16	50	50	0.24
	20	40	0.16	100	20	0.19	20	100	0.13	100	20	0.13
C6	**80**	**80**	**0.20**	**80**	**80**	**0.23**	**100**	**100**	**0.17**	**20**	**140**	**0.35**
	100	100	0.18	100	100	0.19	100	100	0.17	100	100	0.18
	50	50	0.19	50	50	0.22	50	50	0.15	50	50	0.21
	20	20	0.11	100	20	0.16	20	60	0.11	100	20	0.12
RF	**20**	**60**	**0.24**	**20**	**80**	**0.36**	**20**	**20**	**0.25**	**20**	**100**	**0.44**
	100	100	0.20	100	100	0.22	100	100	0.15	100	100	0.15
	50	50	0.24	50	50	0.29	50	50	0.17	50	50	0.21
	100	20	0.18	100	20	0.20	20	100	0.14	100	20	0.12

The lines in bold represent the optimal values for (p, q) obtained by the model-based optimization procedure. For each feature set the last line shows the (p, q) values with worst performance. The second and the third row are baselines. See text for more details.

$\text{merge}_{\text{opt}-\text{opt}-\text{tree}}$: an optimal merging of the feature sets P1, P2, P3, C3, C4, C5, C6 and RF obtained by best-first search (explained further) using the optimal values for p and q from Table 1.

The last method of merging feature sets $\text{merge}_{\text{opt}-\text{opt}-\text{tree}}$ searches for the best performing feature set combination. This should limit the negative effect of information redundancy in the representation. The method is a top-down best-first search procedure that starts from the best performing individual feature set and iteratively adds a feature set based on how this merging performs. In a first iteration the best feature set is merged with each of P2, P3, C3, C4, C5, C6 and RF. For each new feature set (the merging of feature sets) the cost parameter is re-optimized on the same sub-sample and the same 10 CV procedure used to compute the FN5% value in Table 1. The highest FN5% value is selected and again merged with the sets that are left. The procedure is repeated until there are no more sets to merge. Table 2 summarizes the merging process. The $\text{merge}_{\text{opt}-\text{opt}-\text{tree}}$ was applied on the same data sub-sample as used for the (p, q, C) optimization in Table 1. Both the order in which feature sets are added and the associated performance are shown.

Although most of the discriminative information is extracted using the positional information feature set P1 (P2 for human donor sites), the compositional feature sets allow the SVM to significantly increase prediction performance. The coding potential feature set RF seems not to add much more discriminative information. Table 2 also shows that the information redundancy between feature sets decreases prediction performance and a search for the optimal merging of feature sets, as suggested in this manuscript, seems justified.

Table 3 presents the FN5% results of the 10CV evaluation procedure on the larger data sets of 1250 actual and 50 000 pseudo splice sites. It shows that the $\text{merge}_{\text{opt}-\text{opt}-\text{tree}}$ method consistently outperforms the other merging strategies. As a second choice, the $\text{merge}_{\text{opt}-\text{opt}}$ method shows good results as well and does not require the best-first search for the optimal merging of feature sets.

Table 2. Summary of best-first procedure to merge feature sets

Data	Optimal merging
B1 donors	**P1 (0.70)** **C5 (0.75)** **P2 (0.78)** **C3 (0.79)** **P3 (0.80)** C4 (0.79) RF (0.79) C6 (0.76)
B1 acceptors	**P1 (0.57)** **C3 (0.63)** **C5 (0.67)** **P3 (0.68)** **C4 (0.68)** **P2 (0.69)** RF (0.68) C6 (0.66)
B2hum donors	**P2 (0.56)** **C6 (0.61)** **P1 (0.73)** **C3 (0.73)** **P3 (0.74)** RF (0.72) C5 (0.70) C4 (0.68)
B2hum acceptors	**P1 (0.44)** **C4 (0.58)** **C6 (0.61)** **P2 (0.65)** **RF (0.66)** P3 (0.66) C3 (0.66) C5 (0.66)

For each data set the ordering of the feature sets represents the order in which they were selected during the best-first search procedure. Next to each feature set (between brackets) there is the FN5% ratio obtained using a merging of the feature sets up to the ratio. The feature sets in bold represent the optimal merging of feature sets used for $\text{merge}_{\text{opt}-\text{opt}-\text{tree}}$. For B1 donors this optimal merging contains the feature sets P1, P2, P3, C3 and C5. This combined feature set obtains a 0.8 FN5% ratio for the 10CV procedure.

Table 3. Comparison of different feature set merging strategies

	Arabidopsis (B1)		Humans (B2hum)	
	Donors	Acceptors	Donors	Acceptors
merge_{50-50}	0.34	0.27	0.36	0.25
$\text{merge}_{50-50-C\mathit{best}}$	0.34	0.26	0.31	0.23
$\text{merge}_{100-100}$	0.38	0.27	0.36	0.32
$\text{merge}_{100-100-C\mathit{best}}$	0.40	0.30	0.31	0.27
$\text{merge}_{\text{opt}-\text{opt}}$	0.39	0.32	0.45	0.39
$\text{merge}_{\text{opt}-\text{opt}-C\mathit{best}}$	0.39	0.30	0.37	0.33
$\text{merge}_{\text{opt}-\text{opt}-\text{tree}}$	0.43	0.33	0.47	0.44

For each merging strategy the table shows the FN5% ratio obtained using the 10CV procedure on a set of 1250 actual and 50 000 pseudo splice sites.

Prediction performance

The B1 donor and acceptor prediction models used in SpliceMachine have been induced from the actual and pseudo splice sites in the 1495 *Arabidopsis* genes set using the optimal parameter settings (p, q, C) shown in Table 1 and the feature set mergings presented in Table 2. The obtained donor and acceptor models were then used to annotate the AraSet. We also annotated AraSet using the NetGene2 mail server on September 29, 2004, the latest version of SplicePredictor with Bayesian models, and the latest version of GeneSplicer. For SPL we copied the results from a benchmark study of 1999 (Pavy *et al*., 1999) as the current version of SPL is only available as a web-demo. Table 4 shows how SpliceMachine significantly outperforms all other systems at all Se levels. At the 90% Se level, the Sp rate increased from 48 to 62% for donor prediction compared to NetGene2, which was the next best performing system at this Se level. For acceptor sites the Sp rate increased from 42 to 60% compared to NetGene2 at the 84% Se level. For donor site prediction this means that the number of false postive predictions decreased by 43%, for acceptor sites this is 52%.

In B2 an evaluation against the system GeneSplicer was computed using 5CV on the set of 1323 *Arabidopsis* genes and the 1115 human genes. The Se and FP% values reported in Pertea *et al*. (2001) are copied into Table 5 next to the results obtained using SpliceMachine. Again we observe a significant increase in prediction performance for both donor and acceptor data sets. At a 95% Se rate, the FP% rate decreased from 6.4 to 2.2% for human donor sites and from 5.8 to 2.9% for human acceptor sites. For *Arabidopsis* the FP% was decreased from 2.8 to 2.1% for donor and from 4.9 to 2.7% for acceptor site prediction, both at the 95% Se level. This again is a significant reduction in false positive predictions.

The B2hum data set was also used in Chuang *et al*. (2001) to evaluate a Winnow-based (Roth, 1998) splice site prediction system that uses a somewhat similar approach to SpliceMachine, but without the context size optimization. These results are also copied in Table 5. Although SpliceMachine only performs slightly better than Winnow for donor site prediction, the differences are again significant for acceptor site prediction.

Recent publications on computational splice site recognition focus on the dependencies between nucleotide positions in close proximity to the GT or AG dinucleotide in human splice sites (Yeo *et al*., 2003; Castelo *et al*., 2004). The context sizes used in these methods are small (p and q are smaller than 20). To compare this to our large context approach we used the *scoresplice webserver*[1] to annotate the B2hum data set. Table 5 shows the results in the *Maxent* column. As the small context size models basically only capture the position-dependent information, their performance is clearly worse, especially for acceptor prediction.

CONCLUSION

SpliceMachine recognizes splice sites based on the positional, compositional and codon bias information that is extracted from a large local context around each candidate splice site. At the heart of SpliceMachine lies an LSVM model that is fast in both computing the classifier as well as in classifying candidate sites. We have shown that this approach performs significantly better than current state-of-the-art splice site prediction tools used by researchers in the field of molecular biology. The approach also allows for easy incorporation of other types of information such as the presence or absence of certain structural characteristics (Patterson *et al*., 2002) or a branch point motif (Tolstrup *et al*., 1997). The use of binary features facilitates the interpretation of the discriminant function, and future work includes the application of advanced feature subset selection methods to separate the relevant from the irrelevant features. By making the software trainable researchers can evaluate SpliceMachine against other methods on their data.

[1] http://www.genes.mit.edu/burgelab/maxent/Xmaxentscan_scoreseq.html

Table 4. SpliceMachine prediction performance on the set of *Arabidopsis* genes in AraSet

	AraSet donors				AraSet acceptors			
	Se	Sp	Sp SpliceMachine	Sp GeneSplicer	Se	Sp	Sp SpliceMachine	Sp GeneSplicer
NetGene2 all sites	0.94	0.33	0.52	0.30	0.84	0.42	0.60	0.37
NetGene2 score $\geq$0.90	0.90	0.48	0.62	0.43	0.66	0.61	0.76	0.55
NetGene2 score $\geq$0.95	0.80	0.60	0.72	0.59	0.48	0.73	0.85	0.71
NetGene2 score $\geq$0.98	0.61	0.66	0.83	0.77	0.22	0.8	0.92	0.84
NetGene2 score $=1$	0.54	0.70	0.84	0.79	0.21	0.79	0.92	0.84
SPL	0.84	0.30	0.70	0.53	0.76	0.23	0.70	0.46
SplicePredictor 100% learning set	0.94	0.21	0.52	0.30	0.95	0.23	0.30	0.20
SplicePredictor/tau maximal/star-value 14	0.39	0.58	0.87	0.85	0.20	0.52	0.91	0.84
SplicePredictor/tau maximal/star-value 11	0.72	0.44	0.78	0.68	0.56	0.42	0.81	0.65
SplicePredictor/tau maximal/star-value 8	0.92	0.31	0.58	0.37	0.90	0.36	0.47	0.26
SplicePredictor/tau maximal/star-value 5	0.94	0.21	0.52	0.30	0.95	0.25	0.30	0.20

Prediction performance is measured in terms of sensitivity (Se) and specificity (Sp). Each line shows the name of the system, the Se and Sp results of this system on AraSet and (in the SP SpliceMachine column) the Sp value obtained by SpliceMachine at the same Se ratio.

Table 5. Prediction performance of GeneSplicer, SpliceMachine, Maxent and Winnow on the B2ara and B2hum data set described in the text

	Se	FP% Donors				Acceptors			
		GeneSplicer	SpliceMachine	Maxent	Winnow	GeneSplicer	SpliceMachine	Maxent	Winnow
B2ath	0.97	0.047	0.032	—	—	0.117	0.047	—	—
	0.95	0.028	0.021	—	—	0.049	0.027	—	—
	0.93	0.019	0.015	—	—	0.033	0.018	—	—
	0.92	0.017	0.013	—	—	0.029	0.016	—	—
	0.90	0.014	0.010	—	—	0.024	0.012	—	—
	0.85	0.009	0.006	—	—	0.016	0.008	—	—
	0.80	0.006	0.004	—	—	0.011	0.005	—	—
	0.70	0.004	0.002	—	—	0.007	0.003	—	—
B2hum	0.97	0.147	0.032	0.101	0.041	0.093	0.048	0.141	0.078
	0.95	0.064	0.022	0.075	0.030	0.058	0.029	0.107	0.051
	0.93	0.048	0.016	0.059	0.022	0.047	0.021	0.082	0.038
	0.92	0.041	0.014	0.053	0.020	0.043	0.019	0.074	0.034
	0.90	0.035	0.011	0.045	0.016	0.037	0.015	0.061	0.027
	0.85	0.025	0.006	0.033	0.011	0.026	0.009	0.044	0.017
	0.80	0.018	0.004	0.025	0.008	0.019	0.006	0.036	0.012
	0.70	0.007	0.002	0.016	0.003	0.008	0.003	0.022	0.004

The Sp ratios obtained using a 5CV procedure are shown for each Se ratio. The results for GeneSplicer are copied from Pertea *et al*. (2001), the results for Winnow from Chuang *et al*. (2001). The result for Maxent were obtained by submitting the B2ara and B2hum data sets to the *scoresplice webserver*.

REFERENCES

Boser,B., Guyon,I. and Vapnik,V.N. (1992) A training algorithm for optimal margin classifiers. In: *Proc. COLT* (Haussler,D., ed.), ACN Press, Pittsburgh, PA, 144–152.

Cai,D., Delcher,A., Kao,B. and Kasif,S. (2000) Modeling splice sites with Bayes networks. *Bioinformatics*, **16**, 152–158.

Castelo,R. and Guigo,R. (2004) Splice site identification by idlBNs. *Bioinformatics*, **20**, 169–176.

Chuang,J.S. and Roth,D. (2001) Splice site prediction using a sparse network of winnows. Technical Report, University of Illinois, Urbana-Champaign.

Dash,D. and Gopalakrishnan,V. (2001) Modeling DNA splice regions by learning Bayesian networks. Technical report, Center for Biomedical Informatics, University of Pittsburgh.

Degroeve,S., De Baets,B., Van de Peer,Y. and Rouzé,P. (2002) Feature subset selection for splice site prediction. *Bioinformatics*, **18**, 75–82.

Florea,L., Hartzell,G., Zhang,Z., Rubin,G.M. and Miller,W. (1998) A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Res.*, **8**, 967–974.

Hebsgaard,M.S., Korning,G.P., Tolstrup,N., Engelbrecht,J., Rouzé,P. and Brunak,S. (1996) Splice site prediction in Arabidopsis thaliana DNA by combining local and global sequence information. *Nucleic Acids Res.*, **24**, 3439–3452.

Kohavi,R. and John,G. (1997) Wrappers for feature subset selection. *Artificial Intelligence*, **97**, 273–324.

Lim,L.P. and Burge,C.B. (2001) A computational analysis of sequence features involved in recognition of short introns. *Proc. Natl Acad. Sci. USA*, **98**, 11193–11198.

Mathé,C., Sagot,F.M., Schiex,T. and Rouzé,P. (2002) Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids Res.*, **30**, 4103–4117.

Patel,A. and Steitz,L. (2003) Splicing double: insights from the second spliceosome. *Natl. Rev. Mol. Cell Bio.*, **4**, 960–970.

Patterson,D.J., Yasuhara,K. and Ruzzo,W.L. (2002) Pre-mRNA secondary structure prediction aids splice site prediction. *Proceedings of the Pacific Symposium on Biocomputing*. Lihue, Hawaii, World Scientific Press, pp. 223–234.

Pavy,N., Rombauts,S., Déhais,P., Mathé,C., Ramana,D.V.V., Leroy,P. and Rouzé,P. (1999) Evaluation of gene prediction software using a genomic data set: application to Arabidopsis thaliana sequences. *Bioinformatics*, **15**, 887–899.

Pertea,M., Lin,X. and Salzberg,S.L. (2001) GeneSplicer: a new computational method for splice site prediction. *Nucleic Acids Res.*, **29**, 1185–1190.

Reese,G.M., Eeckman,H.F., Kulp,D. and Haussler,D. (1997) Improved splice site detection in genie. *J. Comp. Biol.*, **4**, 311–323.

Roth,D. (1998) Learning to resolve natural language ambiguities: A unified approach. *Proceedings of the National Conference of Artificial Intelligence*. Madison, WI, American Association for Artificial Intelligence, pp. 806–813.

Sonnenburg,S., Ratsch,G., Jagota,A. and Muller,R.K. (2002) New methods for splice site recognition. *Proceedings of In ICANN'02*. Madrid, Spain.

Sun,Y.F., Fan,D.X. and Li,Y.D. (2002) Identifying splicing sites in eukaryotic RNA: support vector machine approach. *Comp. Bio. Med.*, **33**, 17–29.

Tolstrup,N., Rouzé,P. and Brunak,S. (1997) A branch point consensus from Arabidopsis found by non-circular analysis allows for better prediction of acceptor sites. *Nucleic Acids Res.*, **25**, 3159–3163.

Usuka,J. and Brendel,V. (2000) Gene structure prediction by spliced alignment of genomic DNA with protein sequences: Increased accuracy by differential splice site scoring. *J. Mol. Biol.*, **297**, 1075–1085.

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

Yeo,G. and Burge,C. (2003) Maximum entropy modeling of short sequence motifs with applications to RNA splicing signals. *Proceedings of the Seventh International Conference on Research in Computational Molecular Biology*. Berlin, Germany, ACM Press, pp. 322–331.

Zhang,M.Q. and Marr,T.G. (1993) A weight array model for splicing signal analysis. *Comp. Appl. Biosci.*, **9**, 499–509.