BIOINFORMATICS

Vol. 18 Suppl. 2 2002
Pages S75–S83

Feature subset selection for splice site prediction

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

[1]Department of Plant Systems Biology, Flanders Interuniversity Institute for Biotechnology (VIB), K.L. Ledeganckstraat 35, Gent, 9000, Belgium, [2]Department of Applied Mathematics, Biometrics and Process Control, Universiteit Gent, Coupure links 653, Gent, 9000, Belgium and [3]Laboratoire associé de l'INRA (France), K.L. Ledeganckstraat 35, Gent, 9000, Belgium

Received on April 8, 2002; accepted on June 15, 2002

ABSTRACT

Motivation: The large amount of available annotated *Arabidopsis thaliana* sequences allows the induction of splice site prediction models with supervised learning algorithms (see Haussler (1998) for a review and references). These algorithms need information sources or *features* from which the models can be computed. For splice site prediction, the features we consider in this study are the presence or absence of certain nucleotides in close proximity to the splice site. Since it is not known how many and which nucleotides are relevant for splice site prediction, the set of features is chosen large enough such that the probability that all relevant information sources are in the set is very high. Using only those features that are relevant for constructing a splice site prediction system might improve the system and might also provide us with useful biological knowledge. Using fewer features will of course also improve the prediction speed of the system.

Results: A wrapper-based feature subset selection algorithm using a *support vector machine* or a *naive Bayes* prediction method was evaluated against the traditional method for selecting features relevant for splice site prediction. Our results show that this wrapper approach selects features that improve the performance against the use of all features and against the use of the features selected by the traditional method.

Availability: The data and additional interactive graphs on the selected feature subsets are available at http://www.psb.rug.ac.be/gps.

Contact: svgro@gengenp.rug.ac.be; yvdp@gengenp.rug.ac.be

INTRODUCTION

State-of-the-art gene finding systems are integrated models (Haussler, 1998) in which the task of identifying regions that encode proteins in a raw DNA-sequence is divided into a number of subtasks such as the prediction of promotors, start and stop codons, splice sites and other signals and content sensors. Systems that handle these subtasks are then integrated in a more global framework that combines the predictions of the individual models to predict the global structure of a gene.

In *Arabidopsis thaliana*, and practically all other higher eukaryotic organisms, a gene is not a continuous sequence in the DNA, but usually consists of a set of coding fragments known as exons that are separated by non-coding intervening fragments known as introns.

Most introns follow the GT-AG consensus: they start with the GT *consensus dinucleotide* (at the 5'boundary) called a *donor* site and end with the AG *consensus dinucleotide* (at the 3'boundary) called an *acceptor* site. The automatic prediction of these donor and acceptor sites is a crucial step in the gene finding process known as *splice site prediction*.

Two popular methods for the induction of splice site prediction models are the *Weight Matrix Model* (WMM) of Staden (1984) and the more complex *Weight Array Matrix* (WAM) of Zhang and Marr (1993). They are important components in gene prediction systems such as GeneSplicer (Pertea *et al.*, 2001) and GenScan (Burge and Karlin, 1997). Inducing a WMM or WAM model implies the construction of a task-specific data set, which requires the selection of information sources that are considered to be relevant for the task at hand. In our splice site prediction task, these features are typically adjacent nucleotides at fixed positions relative to the candidate splice site, i.e. p adjacent positions upstream and q adjacent positions downstream the candidate.

Optimal values for p and q are obtained by a simple trial and error algorithm where some values for p and q are tested and the best values are selected. We explore a more intelligent search for relevant features, known as *wrapper-based feature subset selection* (Kohavi *et al.*, 1997). This approach uses the predictive performance of an internal prediction model to decide which feature(s) to eliminate.

The choice of the prediction model is a parameter of the wrapper algorithm.

The models we consider in the wrapper algorithm are the WMM and the *Support Vector Machine* (SVM) (Boser *et al.*, 1992; Vapnik, 1995). The latter was selected because of its implicit feature weighing mechanism, which can be well exploited by integrating a feature subset selection method (Mukherjee, 2000; Chapelle *et al.*, 2000; Brown *et al.*, 1999).

In this study, we combine the SVM with feature subset selection in a wrapper algorithm. This approach was applied successfully by Guyon *et al.* (2000) for predicting relevant genes in a microarray experiment. We present a slightly modified version of their feature subset selection method and apply it to select relevant, not necessarily adjacent, nucleotide positions for splice site prediction.

Our modification of the method described by Guyon *et al.* (2000) allowed us to use the same approach for the WMM method. The use of probabilistic models for feature subset selection has been explored by e.g. Langley *et al.* (1994), where the WMM is known as the naive Bayes method (NBM) (Duda and Hart, 1973), which is, as explained further, the exact same method.

METHODS

The splice site data sets

The *Arabidopsis thaliana* data set was generated by aligning mRNAs (with SIM4; Florea *et al.* (1998)) obtained from the public EMBL database (June 5th 2000), with the BAC-sequences that were used for the Arabidopsis chromosome assembly. For future evaluation purposes, we excluded all genes that where in *AraSet* (Pavy *et al.*, 1999). Redundant genes were excluded by applying algorithm 2 of Hobohm *et al.* (1992). This method counts the neighbours (two genes are neighbours when they show more than 80% identity at the nucleotide level) of every gene, discards the gene with the largest number of neighbours and repeats this process until no genes with neighbours 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 positive instances. All GT dinucleotides at the start of these introns are positive donor instances and all AG dinucleotides at the end of these introns are positive acceptor instances. The negative donor instances are defined as, for all genes, all GT dinucleotides that are located between 100 nucleotide positions upstream of the first donor and 100 nucleotide positions downstream of the last acceptor in that gene and that are not donor sites. The negative acceptor instances are defined as all AG dinucleotides within the same range and that are not acceptor sites.

Each positive or negative instance is described by 400 binary features where each feature indicates the presence or absence of one of the 4 nucleotides at a certain position relative to the consensus, i.e. each instance has 100 features set to 1 and 300 features set to 0. We used p nucleotide positions upstream and q nucleotide positions downstream the consensus where $p = q = 50$ resulting in $4 \times (50 + 50) = 400$ binary features. A training data set with balanced class distribution was compiled by random selection of 1000 positive instances and 1000 negative instances (GT_{2000} and AG_{2000}). For the test data set we extracted all candidate splice sites within the interval as defined above from 50 genes. This results in a test data set GT_{50} with 281 positive and 7505 negative instances and a test data set AG_{50} with 281 positive and 7643 negative instances.

The classification methods

As described in the previous section, our data sets contain positive and negative instances that are described by q nucleotide positions downstream and p nucleotide positions upstream the consensus. Formally, a data set T contains l instances $\mathbf{x_i}$ $(i = 1, \ldots, l)$ with each $\mathbf{x_i}$ labelled as y^+ or y^- (known as *classes*), indicating a positive or negative instance respectively. Each index x_{ij} $(j = 1, \ldots, n)$ in vector $\mathbf{x_i}$ is a feature F_j, so

$$\mathbf{x_i} = (F_1, F_2, F_3, \ldots, F_{4(p+q)}). \quad (1)$$

Two methods for discriminating between positive and negative instances are described below. They are supervised classification methods that induce a decision function from the instances in T which can then be used to classify a new instance $\mathbf{z}$ not seen in T.

Support Vector Machines. The Support Vector Machine (SVM) Boser *et al.* (1992); Vapnik (1995) is a data-driven method for solving two-class classification tasks. The Linear SVM (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 a 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

$$pred(\mathbf{z}) = \mathrm{sgn}(\mathbf{w} * \mathbf{z} + b). \quad (2)$$

The hyperplane is computed by maximizing a vector of Lagrange multipliers α in

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

$$\text{constrained to: } 0 \leq \alpha_i \leq C \text{ and } \sum_{i=1}^{l} \alpha_i \, y_i = 0, \quad (3)$$

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).

Function K is a kernel function and maps the features in T, called the input space, into a feature space defined by K in which then a linear class separation is performed. For the LSVM this mapping is a linear mapping:

$$K(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{x_i} * \mathbf{x_j}. \quad (4)$$

The non-linear mappings used in this paper are the Polynomial-SVM (PSVM):

$$K(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i} * \mathbf{x_j} + 1)^d, \quad (5)$$

and the Gaussian-SVM (GSVM)

$$K(\mathbf{x_i}, \mathbf{x_j}) = e^{-|\mathbf{x_i}-\mathbf{x_j}|^2/g^2}. \quad (6)$$

After calculating the α_i's in (3), the decision function (2) becomes:

$$pred(\mathbf{z}) = \text{sgn}\left(\sum_{i=1}^{l} \alpha_i \, y_i \, K(\mathbf{x_i}, \mathbf{z}) + b\right). \quad (7)$$

For the LSVM this function reduces to (2) with

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i \, \mathbf{x_i} \, y_i \, . \quad (8)$$

In (7) each α_i is associated with $\mathbf{x_i}$. After optimizing (3) many α_i's will become zero and the corresponding $\mathbf{x_i}$ will not be used in the decision function (7). All $\mathbf{x_i}$ for which the α_i is not zero are called the support vectors. Typically the size of the set of support vectors is much smaller than l.

Naive Bayes. The NBM Duda and Hart (1973) follows the Bayes optimal decision rule, that tells us to assign a class y^c (c in {+,-}) to an unseen instance $\mathbf{z}$ with features $(F_1^z, F_2^z, \ldots, F_n^z)$ that maximizes $P(y^c|F_1^z, \ldots, F_n^z)$, or the probability of the class y^c given the features $(F_1^z, F_2^z, \ldots, F_n^z)$. By using Bayes' rule we can write $pred(\mathbf{z}) = y^c$ as:

$$y^c = \text{argmax}_c \frac{P(F_1^z, \ldots, F_n^z|y^c) \times P(y^c)}{P(F_1^z, \ldots, F_n^z)} \quad (9)$$

The naive Bayes method then simplifies the problem of estimating $P(F_1^z ... F_n^z|y^c)$ by making the arguable naive independence assumption that the probability of the features given the class is the product of the probabilities of the individual features given the class:

$$P(F_1^z, \ldots, F_n^z|y^c) = \prod_{1 \leq j \leq n} P(F_j^z|y^c). \quad (10)$$

Each probability on the right-hand side can now be estimated directly from the instances in T. The probabilities computed in (9) are the same as the ones obtained by a WMM. The computation for $c = +$ is same as constructing a WMM on the positive instances (splice sites) and $c = -$ in (9) is identical to constructing a background WMM on the negative instances (non-splice sites). Equation (10) is generalized to the WAM model by incorporating correlations between adjacent features:

$$P(F_1^z, \ldots, F_n^z|y^c) = P(F_1^z) \prod_{2 \leq j \leq n} P(F_{j-1}^z, F_j^z|y^c). \quad (11)$$

The feature subset selection methods

For selecting an optimal subset of features from $\{F_1, F_2, \ldots, F_n\}$, given the instances in T, one needs to define what is meant by 'optimal subset of feature' (referred to as the *selection criterion*), and define a *search algorithm* to search for this optimal subset of features in the space of feature subset candidates. Since the number of feature subset candidates is exponential in n and in our experiments n is relatively large, the search algorithm needs to be greedy.

Search Algorithm. A review of different search algorithms can be found in Kohavi *et al.* (1997). Both the SVM and NBM are able to perform well in high-dimensional input spaces, which is confirmed by this study. This allows us to start the search algorithm with the subset containing all features. This subset is known to be a good point to start our search in the space of feature subset candidates. The candidate space is explored with just one operator which eliminates a feature from the current subset. This bottom-up search procedure is called a backward feature elimination (BFE) procedure Kohavi *et al.* (1997) and is greedy enough for our data sets. The BFE procedure takes as input a data set T with a non-sorted list of features F. It returns a sorted list R of features in which the first feature is the least relevant and the last feature is the most relevant feature for solving the classification task. The BFE procedure can be written as the following while-loop:

while (F not empty)
{
model = $induce(T)$
$F_{sel} = selectFeature(T, model)$
add F_{sel} as last element to R
remove F_{sel} from all $\mathbf{x_i}$ in T
}

The $induce(T)$ procedure represents a classification method trained on T and returns a model. The

Table 1. Generalization performance of the classification methods The top header shows the names of the data sets: GT/AG represents a set of donor/acceptor instances, GT_{fss}/AG_{fss} represents a set of donor/acceptor instances with optimal set of features. This set of features differs for each of the prediction systems. The number of positive and negative instances (#positives/#negatives) in a data set is indicated next to the data set. For each data set the table presents sensitivity (Se), specificity (Sp) and F-measure (F)

Method	GT_{50} (281/7505)			AG_{50} (281/7643)			$GT_{50\,fss}$ (281/7505)			$AG_{50\,fss}$ (281/7643)		
	S_e	S_p	F	S_e	S_p	F	S_e	S_p	F	S_e	S_p	F
LSVM	96.1	32.8	48.9	95.0	27.6	42.8	97.9	34.6	51.1	93.9	28.5	43.7
PSVM	98.2	34	50.5	96.1	28.6	44.1	97.2	35.5	52	95.7	29.5	45.1
GSVM	97.9	32.3	48.6	96.8	25.9	40.9	98.6	33.3	49.8	96.4	27.6	42.9
NBM	96.8	27.1	42.3	94.3	24.7	39.1	97.5	30.4	46.3	94.7	25.9	40.7
WMM	96.8	27.1	42.3	94.3	24.7	39.1	96.8	27.8	43.2	95.7	25.1	39.8
WAM	95	27.2	42.3	97.1	29.8	45.6	95.7	28.9	44.4	96.1	30.1	45.8

$selectFeature(T, model)$ procedure selects the feature that optimizes the selection criterion based on the model induced in each iteration of the BFE procedure.

Selection Criterion. A simple criterion consists in selecting that feature that decreases the predictive performance of the model the most. For all features F_i still in F we evaluate the generalization performance of the model after removing F_i from all instances in T. We choose to approximate this generalization performance by the accuracy on T while setting F_i for all instances in T to its mean value. Using cross-validation as an approximation would make the BFE impractical for the real world data sets in this paper.

For the SVM $selectFeature(T, model = \alpha)$ becomes:

$$\text{argmax}_{F_k}\left(\sum_{j=1}^{l} y_j \times (\sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x_i^k}, \mathbf{x_j^k}) + b)\right), \quad (12)$$

where $\mathbf{x_i^k}$ is instance $\mathbf{x_i}$ with feature F_k set to its mean value. In (12) the α_i's are kept constant for all F_k, i.e. no retraining is done when considering to eliminate a feature F_k, but instead the previous values for the α_i's are used, reducing the computation time of BFE drastically.

For the NBM, $SelectFeature(T, model)$ becomes:

$$\text{argmax}_{F_k}(\sum_{j=1}^{l} y_j \times y_j^k), \quad (13)$$

with y_j^k defined as:

$$\text{argmax}_c\left(\frac{P(F_1^j, \ldots, F_{k-1}^j, F_{k+1}^j, \ldots, F_n^j | y^c) \times P(y^c)}{P(F_1^j, \ldots, F_{k-1}^j, F_{k+1}^j, \ldots, F_n^j)}\right). \quad (14)$$

We will refer to selection criterion (12) as BFE_{SVM} and to selection criterion (13) as BFE_{NBM}.

RESULTS AND DISCUSSION

Model induction

The parameters that need to be set by the user were optimized on the GT_{2000} and AG_{2000} data sets using *10-fold cross-validation* (10CV) Weiss and Kulikowski (1991) which divides the data set into ten partitions, uses each partition in turn as test set and the other nine as training set, and computes evaluation criteria by averaging the results for the ten test sets. The partitions we used were equally sized, contained a balanced class distribution and were the same for each method. Parameter values that resulted in the best predictive performance were selected. In case of the SVM the number of support vectors in the model was used as a second criterion for parameter selection in case two models would have nearly the same predictive performance.

Columns GT_{50} and AG_{50} of Table 1 show the predictive performance on our test data sets, after training on resp. GT_{2000} and AG_{2000}. In these first two columns, all algorithms were given all of the 400 binary features. Predictive performance is measured in terms of the sensitivity (S_e), defined as the proportion of positives that are correctly predicted, and the specificity (S_p), defined as the proportion of predicted positives that are correct. The F-measure (F) is used to combine sensitivity and specificity in an harmonic mean:

$$F = 2 \times \frac{S_e \times S_p}{S_e + S_p}. \quad (15)$$

Although the aim of this study is not to improve splice site prediction with the SVM method, we do observe a promising improvement in the donor site prediction accuracy.

Feature subset selection

Initially, T in both BFE methods is the $\{GT, AG\}_{2000}$ data set with all features F. In each BFE iteration one

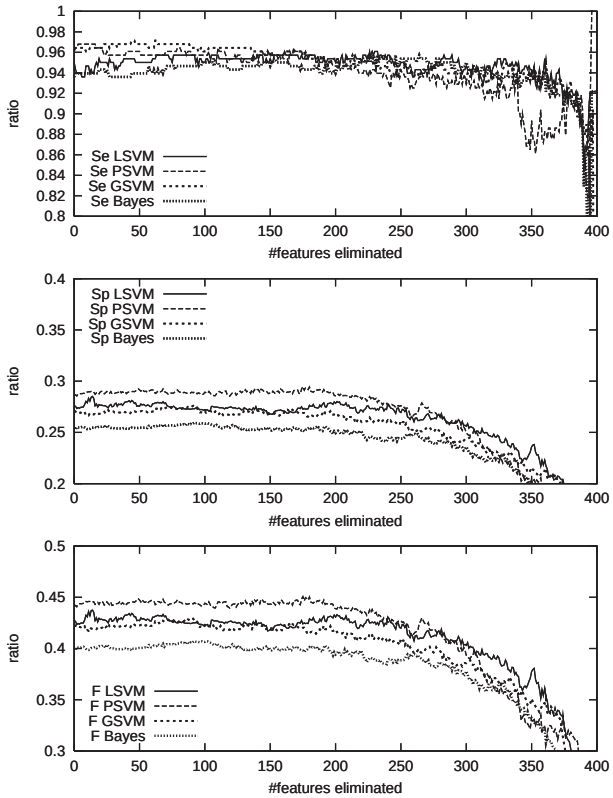

Fig. 1. Feature Subset Selection for donor prediction. In each iteration of BFE a feature is eliminated and the resulting model is evaluated by computing the F-measures (top), sensitivity (middle) and specificity (bottom) ratios on the GT_{10000} data set.

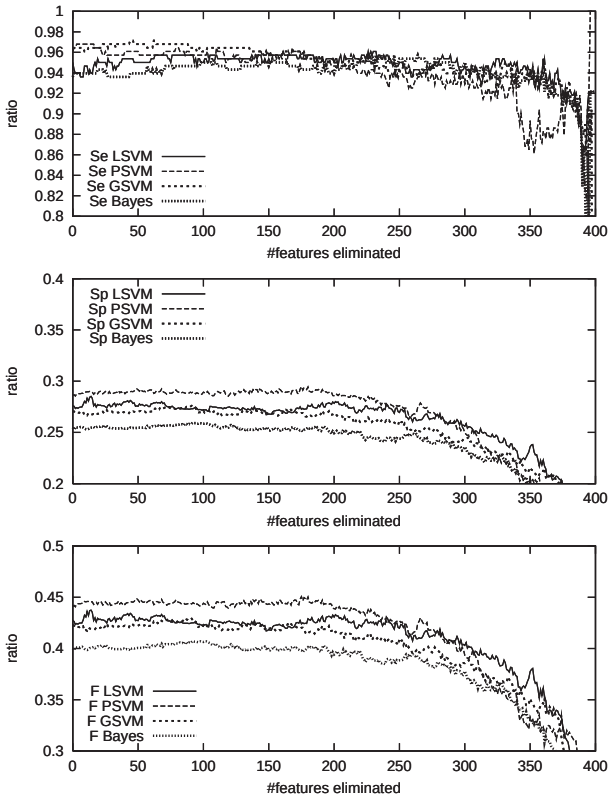

Fig. 2. Feature Subset Selection for acceptor prediction. In each iteration of BFE a feature is eliminated and the resulting model is evaluated by computing the F-measures (top), sensitivity (middle) and specificity (bottom) ratios on the AG_{10000} data set.

feature is removed and the resulting T is evaluated on the $\{\mathsf{GT}, \mathsf{AG}\}_{50}$ data set. Figures 1–4 show the evolution of the predictive performance when eliminating features in a graph where the x-axis is the number of features eliminated so far and the y-axis shows the F-measure and the sensitivity and specificity ratios.

The SVM is not able to compute a predictive model with few features. Computing a unique threshold b (2) becomes impossible which leads to the strange behavior of the $\mathsf{BFE}_{\mathsf{SVM}}$ methods in the very last iterations. What happens is that all instances are either all predicted as positives, or all predicted as negatives, or the return value of the decision function (7) is always zero, which means no prediction at all. The NBM method does not suffer from this limitation and can induce a model with just one feature. For instance, specificity and sensitivity ratios for NBM trained on GT_{2000} with only the presence or absence of nucleotide G at one position upstream the GT consensus are 0.815 and 0.123 respectively. This feature was also selected as most relevant by all three SVM's.

Two conclusions can be drawn from Figures 3 and 4: (a) prediction performance only decreases significantly in the ten last BFE iterations, and (b) prediction performance does increase gradually when adding more features to this ten-features subset. These ten last features in (a) are all very close to the splice site (see also Figures 5–6 described below). It is known that the presence or absence of certain nucleotides at fixed positions close to the splice site (3 to 4 positions upstream and downstream) are relevant for *Arabidopsis* splice site prediction. We see that both SVM and NBM extract this knowledge as most relevant.

Observation (b) can be explained as an attempt to extract position invariant information or features. From Lim and Burge (2001) we learned that the presence of certain pentamers (such as 'TTTTG') in the intron are important discriminating information sources for Arabidopsis splice site prediction. The position of these words relative to the splice site is of less importance, which makes them position invariant features. A classification method that tries to extract such a position invariant feature needs many position dependent features to do so. The more position dependent features are used, the larger the interval

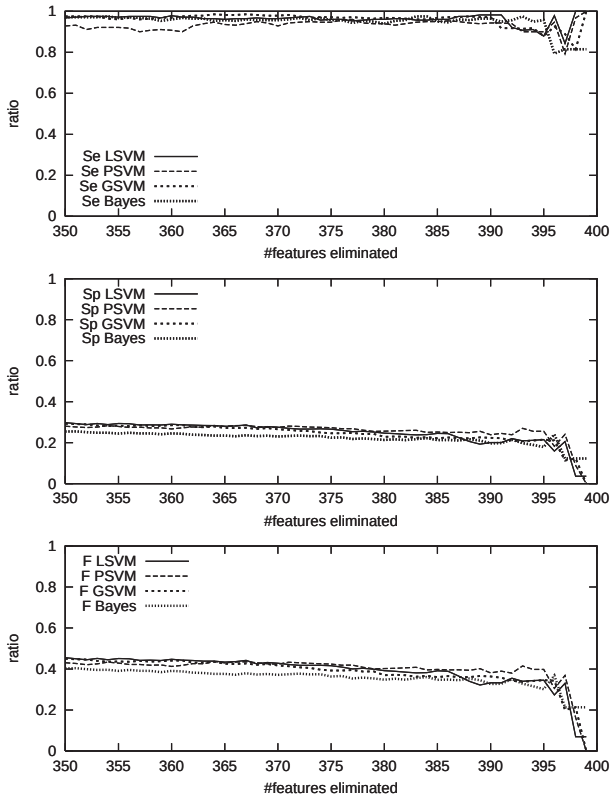

Fig. 3. Last fifty BFE iterations for donor prediction. In each iteration of BFE a feature is eliminated and the resulting model is evaluated by computing the F-measures (top), sensitivity (middle) and specificity (bottom) ratios on the GT_{10000} data set.

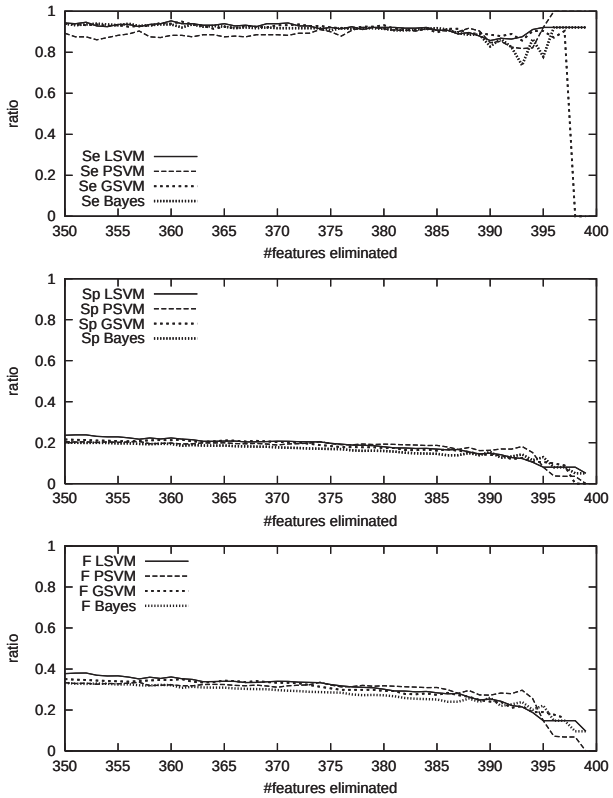

Fig. 4. Last fifty BFE iterations for acceptor prediction. In each iteration of BFE a feature is eliminated and the resulting model is evaluated by computing the F-measures (top), sensitivity (middle) and specificity (bottom) ratios on the AG_{10000} data set.

in which the position invariant feature can be found. We assume that this explains the many adjacent nucleotides T and G in the intron part of the splice site for the 100-features subset (Figures 5–6). It would also explain the gradual increase in predictive performance: the interval in which the position invariant feature can be found becomes at most one position wider when adding one 'relevant' feature to the data set.

Figures 5–6 show snapshots of the BFE feature subset selection process for PSVM (an interactive web page containing all subsets for all four methods will be available at time of publication at http://www.psb.rug.ac.be/gps). Different subsets are shown graphically, together with their performance on $\{\mathsf{GT}, \mathsf{AG}\}_{50}$.

Comparing the feature subsets selected by the four methods, it is difficult to find clear similarities. For an optimal subset containing 10 features, 6 features are selected by all four methods on GT_{2000}. For a subset of 5 features, 4 features where selected by all four methods. On AG_{2000}, the difference between the 10 most relevant features selected by PSVM and the 10 most relevant feature selected by the other three methods is much bigger. Only 3 features where selected by all four methods, but 7 features can be found in the set of 10 most relevant features of the three other methods. The fact that the other features differ might be due to the presence of redundant features. Such a feature can be replaced by another one that is correlated with the replaced feature, i.e. the predictive performance of the system does not change when substituting redundant features.

Columns GT_{fss} and AG_{fss} of Table 1 show the predictive performance on our test data sets, after training on resp. GT_{2000} and AG_{2000}, when using the 'optimal' set of features for each of the four methods used in BFE. The 'optimal' set is defined as the set of features with the best prediction performance, not taking into account the number of features. For the WMM and WAM methods, we used the traditional search for relevant features: all possible combinations of q and p were evaluated and the 'optimal' values were selected. This explains the difference in performance between NBM and WMM in these last two columns.

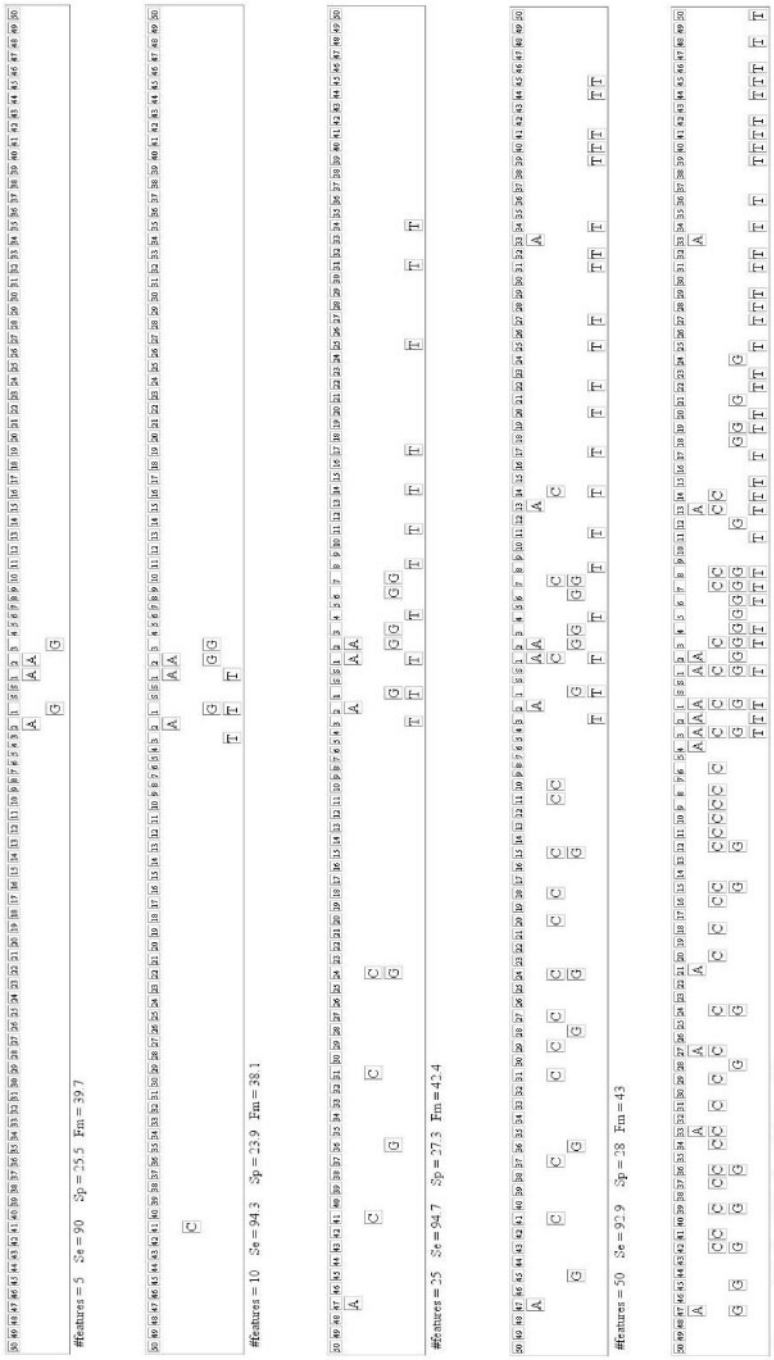

Fig. 5. Snapshots from $\mathsf{BFE}_{\mathsf{PSVM}}$ trained on GT_{2000} and tested on GT_{50}. From bottom to top features are eliminated and F-measure (Fm), sensitivity (Se) end specificity (Sp) ratios are given for PSVM models trained with the features in the box above the ratios.

Removing features with the BFE method clearly improves the prediction performance of all four methods on both GT and AG. Also, compared to the traditional method applied to WMM, the improvement is significantly higher. The F-measure for NBM on GT improves from 42.3 to 46.3, while the WMM improves from 42.3 to 43.2. A comparable observation can be made for the AG data set.

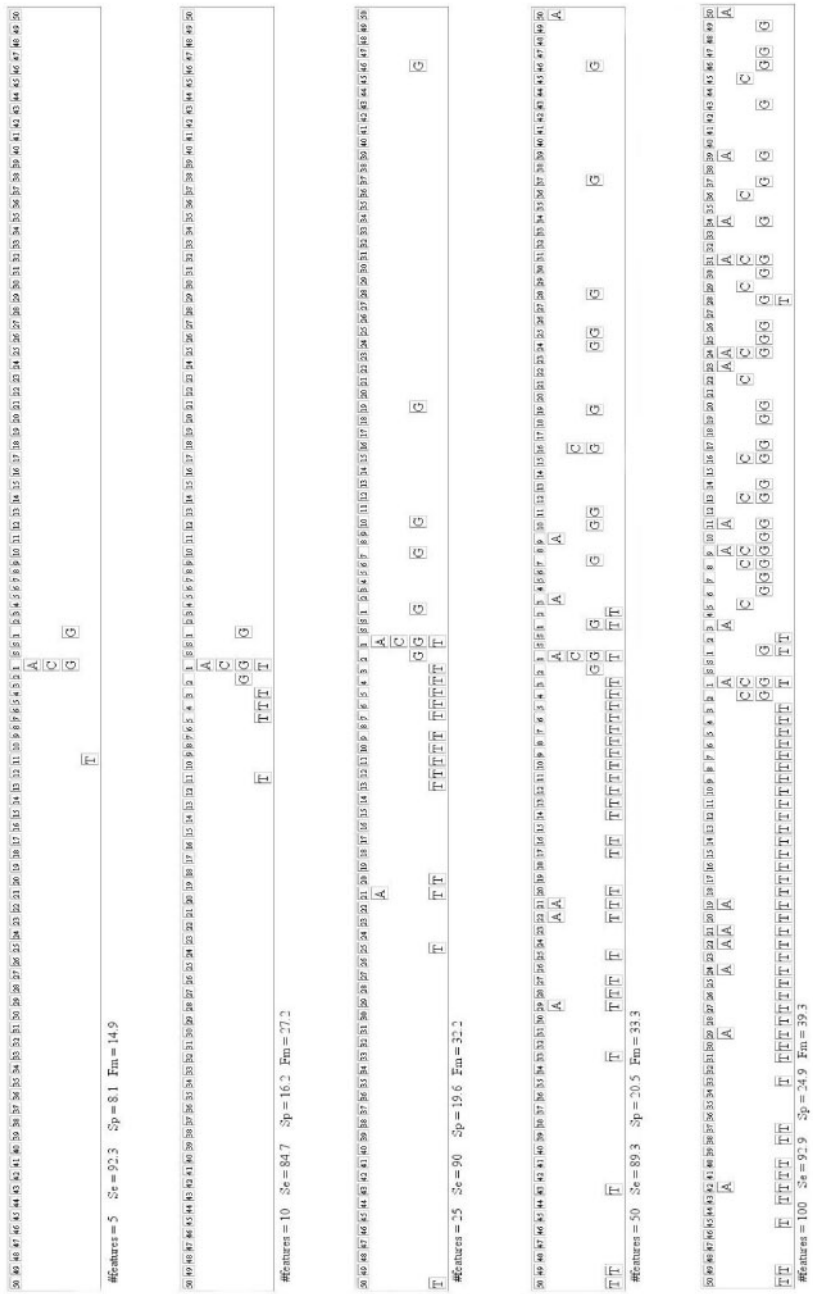

Fig. 6. Snapshots from $\mathsf{BFE}_{\mathsf{PSVM}}$ trained on AG_{2000} and tested on AG_{50}. From bottom to top features are eliminated and F-measure (Fm), sensitivity (Se) end specificity (Sp) ratios are given for PSVM models trained with the features in the box above the ratios.

CONCLUSIONS

We have described how to apply a more advanced BFE search for relevant features for the splice site prediction task and have shown how prediction performance benefits from BFE compared to the traditional approach.

Looking at the selected feature subsets, we conclude that the only position dependent nucleotides that are relevant for splice site prediction are those that are in close proximity to the splice site. Both the SVM and the NBM try to extract and use position invariant features such as e.g. the pentamers discussed in Lim and Burge (2001) by using a set of position dependent features.

A classification method that uses position dependent features should use only those nucleotides that are close to the splice site. The gain in predictive performance that is observed when using more position dependent features should be compensated by adding position invariant features such as covariation, oligomers and secondary structure.

ACKNOWLEDGEMENTS

Yves Van de Peer is a Research Fellow of the National Fund for Scientific Research-Flanders. The authors thank Stephane Rombauts for sharing his biological expertise on DNA splicing and for his help with compiling the data sets. Special thanks also to Sébastien Aubourg for compiling the splice site data sets.

REFERENCES

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

Brown,M., Grundy,W., Lin,D., Cristianini,N., Sugnet,C., Ares,M. and Haussler,D. (1999) Support vector machine classification of microarray gene expression data. *University of California, Santa Cruz, Technical Report UCSC-CRL-99-09*.

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

Chapelle,O., Vapnik,V., Bousquet,O. and Mukherjee,S. (2000) Choosing kernel parameters for support vector machines. *AT&T Labs Technical Report. March, 2000*.

Duda,R.O. and Hart,P.E. (1973) *Pattern Classification and Scene Analysis*. Wiley, New York, NY.

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.

Guyon,I., Weston,J., Barnhill,S. and Vapnik,V.N. (2000) Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*.

Haussler,D. (1998) Computational genefinding. *Trends Biochem. Sci.*, 12–15 Suppl. S.

Hobohm,U., Scharf,M., Schneider,R. and Sander,C. (1992) Selection of representative protein data sets. *Protein Sci.*, **1**, 409–417.

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

Langley,P. and Sage,S. (1994) Induction of selective Bayesian classifiers. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kauffman, pp. 399–406.

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

Mukherjee,S., Tamayo,P., Slonim,D., Verri,A., Golub,T., Messirov,J.P. and Poggio,T. (2000) Support vector machine classification of microarray data. *AI memo 182., CBCL paper 182., MIT*.

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.

Staden,R. (1984) Computer methods to locate signals in nucleic acid sequences. *Nucleic Acids Res.*, **12**, 505–519.

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

Weiss,S. and Kulikowski,C. (1991) *Computer Systems that Learn*. Morgan Kaufmann, San Mateo, CA.

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