# Learn from the Information Contained in the False Splice Sites as well as in the True Splice Sites using SVM

**Shuo Xu[1] Fujing Ma[2] Lan Tao[3]**

[1] College of Information and Electrical Engineering, China Agricultural University (East Campus), P.O. Box 215, Beijing 100083, P.R. China

[2] Seneca College of Applied Arts and Technology, Shandong Institute of Business and Technology, Yantai 264005, P.R. China

[3] College of Information Engineering, Shenzhen University, Shenzhen 518060, P.R. China

## Abstract

In splice sites prediction, the information contained in false splice sites is often ignored, which has been recognized to be very valuable. In this paper, three novel encoding approaches, MCM with DTF, MCM with UTF and WAM with UTF, are described, all of which consider the information both in true and false splice sites. From the comparison with 5 other encoding methods, we can conclude: (1) SVM can benefit from the information contained in false splice sites as well as in true splice sites. (2) The performance of MCM with DTF and WAM with DTF is comparative, both of which give the better performance nearly in all cases. (3) The performance of binary vector encoding method is surprisingly good, the potential of which need to be further investigated.

**Keywords**: SVM, MCM, WAM, DTF, UTF

## 1. Introduction

Identification of protein coding genes in genomic DNA becomes an increasing important task in bioinformatics, often referred to as gene finding. For most eukaryotic genomes, a protein coding gene consists of a set of regions called exons, usually separated by other regions called introns. The 5' boundary or donor site of these introns usually contains the dinucleotide GT (GU in pre-mRNA), while the 3' boundary or acceptor site contains the dinucleotide AG. The occurrence of splice sites (i.e., donor and acceptor sites) in genomic sequences is an important characteristic for gene finding. It has been recognized that accurate prediction of higher eukaryotic gene structure largely depends on the ability to pinpoint the exact splice sites [23].

Through the rapid sequencing of genes and their cognate transcripts, the number of experimentally confirmed splice sites has grown extensively, which makes it possible to predict splice sites based on machine learning approaches. Splice sites prediction can be divided into two subtasks: donor sites prediction and acceptor sites prediction, either of which can be formally stated as a binary classification problem: {donor site, non-donor site} and {acceptor site, non-acceptor site}. A number of computational methods have been developed to identify these splice sites, including stand-alone splice site finders and gene finders. Most of the latter have a modular structure, in which splice site predictor is a critical component [19].

The Markov chain model (MCM) is a well-known tool for analyzing biological sequence data. Moreover, higher order MCMs are often considered to be more efficient in capturing possible interactions among nucleotides surrounding the splice sites [15]. However, with increasing order of the MCM, the number of model parameters increases exponentially (see further), which makes it impossible to obtain a stable estimation of parameters with the limited amount of training data. Because in order to estimate parameters of a $k$-order MCM, many occurrences of all possible ($k$ + 1)-mers must be appear in the training sequences. In other words, their direct implementations are practically prohibitive. However, in order to take advantage of the strengths of higher order MCMs, several approximation techniques and algorithms have so far been developed. Salzberg et al. [9]-[10] proposed an IMM (Interpolated Markov Model) approach, which utilized a linear combination of probabilities obtained from several lengths of oligomers (i.e., lower order MCMs) to make predictions, giving high weights to oligomers that occur frequently and low weighs to those that do not. Ohler and Reese [20] and Ohler et al. [21] put forward another IMM approach. NN (Neural Network) and SVM (Support Vector Machine), both of which are very effective methods for general-purpose pattern

recognition, are capable of learning complex interactions of nucleotides by finding arbitrarily complex non-linear mapping. Ho and Rajapakse [17] and Baten et al. [4] have shown respectively that the higher order MCMs can be approached by NN and SVM with taking the outputs of low order MCMs as inputs.

The SVM method was proposed initially by Vapnik and his co-workers [11]-[12], which is not only well-founded theoretically, but also has a number of interesting properties, including effective avoidance of overfitting and underfitting, the ability to handle large feature spaces, information condensing of the given data set, etc. Shortly after its introduction, its performance has already either matched or outperformed that of traditional machine learning approaches (e.g., NN) for a wide range of applications including splice sites prediction [2]-[7]. Currently, the SVM approach mainly deals with numerical data (with the exception of special kernel functions), so the DNA sequences must be encoded beforehand in some way. Thus the low order MCMs can also be viewed as pre-processing step for the SVM. In addition, since there are two parts in the training data set: true splice sites and false splice sites, both of which are known prior. However, the information contained in false splice sites is often ignored, which has been recognized to be very valuable [1]-[2]. We are interested in how to make use of the information contained in the false splice sites as well as in the true splice sites. Motivated by [17] and [2], we accordingly design three novel encoding approaches: MCM with DTF, MCM with UTF, and WAM with UTF, all of which consider the information both in true and false splice sites.

## 2. Markov chain model

A MCM can be seen as a collection of states, in which observed state variables are drawn from the alphabet $\Omega_{DNA} = \{A, C, G, T\}$. The number of states is equal to the length of each sequence, and each state variable of the model corresponds to a nucleotide in the sequence. For convenience, let us consider a sequence of length $l$: $s = (s_1, s_2, \ldots, s_l)$, where $s_i \in \Omega_{DNA}$, $i = 1, \ldots, l$. Then the nucleotide $s_i$ is a realization of the $i$ th observed state variable of the MCM, and only state transitions from state $i$ to state $i + 1$ are allowed. It emits symbols from the alphabet $\Omega_{DNA}$ at each state, and then serially evolves form one state to the next state, where the emission probability of each symbol at each state is characterized by a position-specific probability parameter. Assume the MCM is of $k$ th order, then the likelihood of the sequence $s$ generated by the model $M$ is

$$P(s_1, s_2, \cdots, s_l | M) = P(s_1, \cdots, s_k | M) \prod_{i=k+1}^{l} P(s_i | s_{i-k}, \cdots, s_{i-1}, M) \quad (1)$$

where $P(s_1, \ldots, s_k \mid M)$ is the independent probabilities of the oligomers of length $k$: $(s_1, s_2, \ldots, s_k)$, and $P(s_i \mid s_{i-1}, s_{i-2}, \ldots, s_{i-k}, M)$ denotes the conditional probability of emitting a nucleotide at position $i$ given $k$ previous ones. Given $n$ aligned sequences of length $l$ (i.e., training data), the maximum likelihood (ML) estimation of the emission probability can be performed simply by counting the oligomers of length $k + 1$ and $k$ in a set of training sequences:

$$P(s_i | s_{i-k}, \cdots, s_{i-1}, M) = \frac{P(s_{i-k}, \cdots, s_{i-1}, s_i | M)}{P(s_{i-k}, \cdots, s_{i-1} | M)} = \frac{\#(s_{i-k}, \cdots, s_{i-1}, s_i)}{\#(s_{i-k}, \cdots, s_{i-1})} \quad (2)$$

where $\#(\cdot)$ denotes the frequency of its argument in the training samples. That is, a $k$-order MCM can be fully expressed by a set of $4^k + (l - k)\times 4^{k+1}$ parameters:

$$\{P(s_1, \cdots, s_k | M), s_i \in \Omega_{DNA}, i = 1, \cdots, k\} \cup \{P(s_i | s_{i-k}, \cdots, s_{i-1}, M), s_i \in \Omega_{DNA}, i = k+1, \cdots, l\} \quad (3)$$

In this paper, the first order MCM is utilized to model position-specific dependencies among nucleotides surrounding the splice sites. The parameters in the first order MCM can be tabulated in a $16\times l$ matrix, where the first column contains the independent probabilities of the four nucleotides, and all the remaining columns contain conditional probabilities. This matrix is referred to as a MCM matrix.

## 3. Methods of encoding

Two sample logo [24] figures shown in [4] suggest that there is a significant difference between vicinities of the true and false splice sites. That is, there exists some valuable information in false splice sites as well as in true splice sites, which has been pointed out by Yin and Wang [1] and Huang et al. [2] and validated in their computational experiments. However, the information contained in false splice sites is often ignored [4]. In what follows, we proposed three novel methods to make use of the information contained in false splice sites as well as in true splice sites.

### 3.1. MCM with DTF

According to Eq. 3, two MCM donor (resp. acceptor) matrices $M^{\mathrm{T}}$, $M^{\mathrm{F}}$ can be derived from true and false donor (resp. acceptor) sites, respectively. Motivated by Huang et al. [2], an encoding matrix $M$ is obtained by subtracting the false donor (resp. acceptor) sites matrix $M^{\mathrm{F}}$ from the true donor (resp. acceptor) sites matrix $M^{\mathrm{T}}$, i.e., $M = M^{\mathrm{T}} - M^{\mathrm{F}}$. Then the training and testing data are encoded with this encoding matrix $M$. This encoding method is referred to as MCM with difference between the true and false sites (DTF).

## 3.2. MCM with UTF

After two MCM donor (resp. acceptor) matrices $M^{\mathrm{T}}$, $M^{\mathrm{F}}$ are obtained, each sequence in training and testing data set is encoded both with $M^{\mathrm{T}}$ and with $M^{\mathrm{F}}$. Then these two encoded sequences are concatenated as inputs for SVM. This encoding method is indicated as MCM with the union of the true and false sites (UTF).

## 3.3. WAM with UTF

The weight array method (WAM), closely related to first order MCM, was introduced for splice sites prediction by Zhang and Marr [26] on the basis of the weight matrix model (WMM) [22]. However, the WAM as an encoding method for splice sites prediction was first proposed by Baten et al. [4], initially referred to as WMM1. Here, we briefly introduce how to obtain a WAM matrix from the training data. Given $n$ aligned sequences of length $l$, $S_1$, $S_2$, …, $S_n$, where $S_k = (S_{k1}, S_{k2}, \ldots, S_{kl})$, $S_{ki} \in \Omega_{DNA}$, $k = 1, \ldots, n$, $i = 1, \ldots, l$. A $16\times(l-1)$ WAM matrix $M$ is obtained as follows

$$M_{ij} = \frac{1}{n}\sum_{k=1}^{n}\delta_i\left(S_{k,j}, S_{k,j+1}\right), \quad i = AA, AC, \cdots, TT, \; j = 1, \cdots, l-1 \tag{4}$$

$$\delta_i\left(s_1, s_2\right) = \begin{cases} 1, & if \;\; i = s_1 s_2 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Each element in this matrix indicates the number of times that a given dinucleotide is observed at a given position. Likewise, the encoding methods: WAM with DTF and WAM with UTF can be obtained, where the former was put forward by Huang et al. [2], initially called as PN with FDTF.

# 4. Support vector machines

The SVM has a strong theoretical root (Statistical Learning Theory, SLT), the basic idea of which is to represent the sample set with minor support vectors. In essence, training data first are mapped to a high (possibly infinite) dimensional feature space, and then an optimal linear function in this space is constructed according to Structural Risk Minimization (SRM) principle, not Empirical Risk Minimization (ERM) principle as in NN. This allows giving guarantees for the high performance on unseen data, i.e., good generalization ability. The key is a good choice of the so-called kernel function which implicitly defines the feature space and implements dot product operation of feature space in low dimensional input space. In what follows, a brief introduction is given. See [11]-[12] for more details.

In fact, the SVM for binary classification problem can be expressed as a convex quadratic programming problem given by:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) - \sum_{j=1}^{n}\alpha_j \\ s.t. \quad & \sum_{i=1}^{n} y_i \alpha_i = 0 \\ & 0 \le \alpha_i \le C_+, \text{if } y_i = +1 \\ & 0 \le \alpha_i \le C_-, \text{if } y_i = -1 \end{aligned} \tag{6}$$

where $\mathbf{x}_i \in \mathrm{R}^n$ denotes an encoded donor (resp. acceptor) sites training example, $y_i \in \{+1, -1\}$ is the corresponding class, i.e., true (+1) or false (-1) donor (resp. acceptor) site, $n$ is the number of training examples, $K$ is the kernel function, $C_+$ and $C_-$ are the cost we pay when misclassify a true and false donor (resp. acceptor) site, respectively, and $\boldsymbol{\alpha}$ is a vector of Lagrange multipliers that needs to be optimized, each component of which corresponds to one particular training example. After the training process, only a small part of $\alpha_i$s have non-zero values, whose corresponding training examples are called the support vectors.

For splice sites prediction, the number of the false sites is often much more than that of the true sites. Let $n_+$ and $n_-$ are the number of true and false donor (resp. acceptor) site sequences in training data set, respectively, then $n = n_+ + n_-$. In this paper, we set

$$C_+ = \frac{n_-}{n}C \;,\; C_- = \frac{n_+}{n}C \tag{7}$$

where $C$ is a penalty parameter preset by user. In this way, given $C$, we can assign a more cost for misclassification of a true donor (resp. acceptor) site sequence than for misclassification of a false donor (resp. acceptor) site sequence. In this paper, two separate SVM classifiers are constructed, one for donor sites and the other for acceptor sites.

Once the vector of Lagrange multipliers $\boldsymbol{\alpha}$ is known, then the decision function is given by

$$f(\mathbf{x}) = \theta\left(\sum_{i \in SV} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right) \quad (8)$$

where $SV$ is the set of support vectors, and $\theta(x)$ is +1, if x is greater than or equal to a certain threshold, -1 otherwise.

## 5. Material and data

Like most other methods, the SVM also seeks for consensus motifs or features underlying surrounding the splice sites, by learning from sets of training data containing true and false splice sites. This implies that a task-specific data set must be constructed, which requires the selection of information sources that are considered to be relevant for splice sites recognized by a complex of proteins and small nuclear RNAs, known collectively as the spliceosome. However, for now, biochemical details of the mechanism of RNA-splicing have not been understood completely, which limits features extraction from the viewpoint of biochemistry. In general, the selected features are adjacent nucleotides at fixed positions relative to the candidate splice site, i.e., $l_1$ adjacent positions upstream and $l_2$ adjacent positions downstream the candidate splice site as shown in Fig. 1. The total length $l$ is $l_1 + l_2 + 2$.

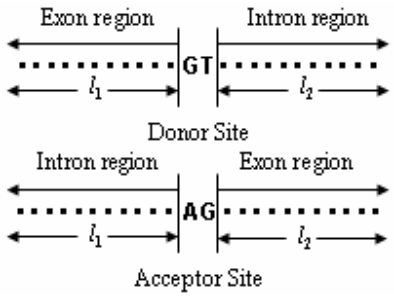


Fig. 1: The length of sequence around candidate donor and acceptor sites.

NN269 data set[1] was collected to develop and test algorithm for human splice sites identification in GENIE system [13]-[14]. It consists of 1324 confirmed true donor sites, 1324 confirmed true acceptor sites, 4922 false donor sites and 5553 false acceptor sites collected from 269 human genes. Each of the false donor/acceptor sites also has GT/AG in the splicing site but is not a real splice site according to the annotation. The window size for a donor is 15 nucleotides ($l_1$ = 7, $l_2$ = 6), for an acceptor 90 nucleotides ($l_1$ = 68, $l_2$ = 20). This data set is split into a training set and a testing set. The training data set contains 1116 true donor, 1116 true acceptor, 4140 false donor and 4672 false acceptor sites. The testing data set contains 208 true donor, 208 true acceptor, 782 false donor and 881 false acceptor sites. However, there is an ambiguity in splice.train-real.D (No.: 903, HSG17G_3388) and splice.train-false.D (No.: 358, HUMA1GLY2_3604), which are excluded from our experiments.

Since the training data is limited, some rare events (positional dinucleotides) may be observed in the testing data but not in the training data. In order to avoid zero probabilities for these events, pseudo counts are introduced. In fact, they have a natural probabilistic interpretation as the parameters of Bayesian Dirichlet prior distributions on the probabilities for each state [16]. Though the pseudo counts should reflect our prior biases about the probability values, for simplicity, the observed frequency of each dinucleotide at each position is increased by 1/16 (the most plain prior knowledge) before used to estimate the corresponding probability parameters in MCM and WAM.

## 6. Performance and assessment

In order to assess prediction performance, receiver operator curve (ROC) analysis is used, which is a graphical representation of sensitivity (*Se*) and specificity (*Sp*) for a binary classification model. In this study, ROC is created from the false positive rate (on the x axis) and the sensitivity (on the y axis). The closer a curve follows the left-hand border and then the top of the border of the ROC plot is, the more accurate the classification model is [25].

The sensitivity, also known as true positive rate (TPR), is the proportion of correct prediction of true sites. The specificity is the proportion of predicted sites that are actually true sites. The false positive rate (FPR) is the proportion of incorrect prediction of false sites. Formally, they are defined as

$$Se = \frac{TP}{TP + FN}, Sp = \frac{TP}{TP + FP}, FPR = \frac{FP}{FP + TN} \quad (9)$$

where $TP$ and $TN$ are the number of the correctly predicted true and false splice sites, respectively, and $FP$ and $FN$ are similarly the number of the incorrectly predicted true and false ones, respectively. By varying the decision threshold used to map Eq. 8 onto a class, $Sp$ and $FPR$ can be calculated for all $Se$ levels.

## 7. Results and discussion

[1] This data set is available at http://www.fruitfly.org/seq_tools/datasets/Human/GENIE_96/splicesets/.

## 7.1. Parameters optimization

For the implementation of SVM, LIBSVM[2] (version 2.83) is utilized, which is a library for SVM. Here, we adopt the radial basis function (RBF) kernel function:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \left|\mathbf{x}_i - \mathbf{x}_j\right|^2\right) \qquad (10)$$

Now there are still two parameters, C [Eq. 7] and γ in the RBF kernel function [Eq. 10], which are unknown beforehand. A two-step grid search [18] is used to select the optimal parameters, and the *Sp* ratio at 5% false negative predictions (*Se* = 0.95) is used as the criterion to measure prediction performances. This measure is referred to as FN5% ratio [3]. First we do a coarse grid search using the following sets of values: $C \in \{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$ and $\gamma \in \{2^{-15}, 2^{-13}, \ldots, 2^{3}\}$. For all possible combinations (C, γ) the FN5% ratio is calculated using 10-fold cross validation. That is, the training data is randomly divided into 10 subsets of nearly equal size while preserving the class distribution. A model is induced 10 times, each time leaving out one of the subsets that is then used to calculate the FN5% ratio. An optimal pair (C*, γ*) is selected from this coarse grid search. In the second step, a fine grid search is conducted around (C*, γ*), with $C \in \{2^{-1.75}\times C^*, 2^{-1.5}\times C^*, \ldots, 2^{1.75}\times C^*\}$ and $\gamma \in \{2^{-1.75}\times\gamma^*, 2^{-1.5}\times\gamma^*, \ldots, 2^{1.75}\times\gamma^*\}$.

The final optimal parameter pair is selected from this fine grid search. In each grid search, especially in the fine grid search step, it is quite often the case that there are several pairs of parameters that give the same 10-fold cross validation FN5% ratio. In this situation, we select the pair with the minimum number of support vectors. In addition, in the coarse grid search the optimal value for C or γ may be at the border of the search space. In this situation the search space for the parameter that is at the border is increased by the same step as described above ($2^{\pm 2}$) until no further improvement is observed.

## 7.2. Performance comparison

In this paper, we totally consider 8 encoding methods: MCM [4], MCM with DTF, MCM with UTF, WAM [4], WAM with DTF [2], WAM with UTF and 4-bit [7]-[8], 16-bit [6] binary vector encoding. MCM and WAM encoding method only consider the information contained in true donor (resp. acceptor) sites. Apart from 4-bit binary vector encoding, the other methods consider explicitly the dependencies between adjacent nucleotide positions. One of reasons that 4-bit binary vector encoding is included is that it is simple and commonly used. For MCM and WAM encoding methods, although Baten et al. [4] reported their performances in NN269 data set, we have re-conducted the experiment with the same setting as in [4], and the performance obtained has a very large gap with theirs. We have contacted with Baten, and he has said that there should not be any problem, but he has promised to check whether he has uploaded the right code. Before we get his response, here we only give our obtained results.

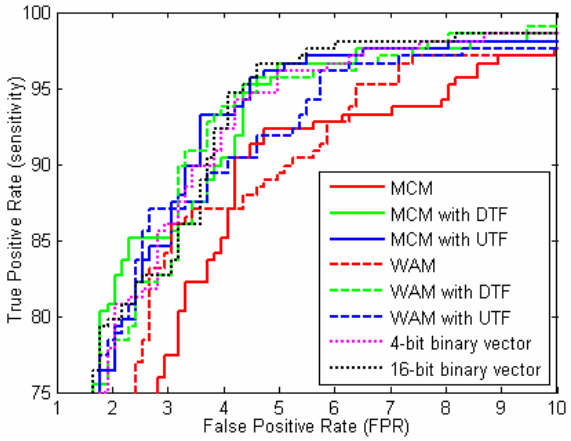


Fig. 2: Comparison of performance for different encoding methods in donor sites prediction.

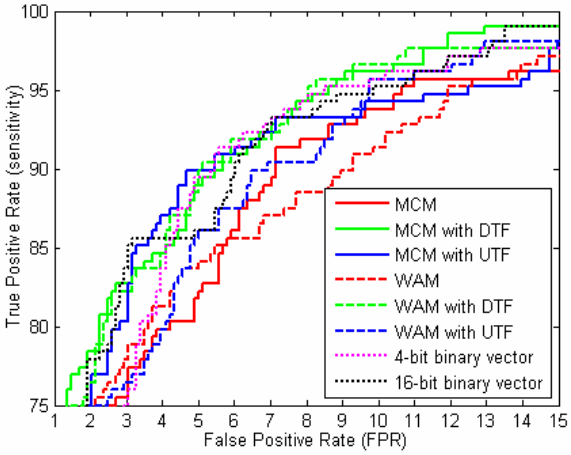


Fig. 3: Comparison of performance for different encoding methods in acceptor sites prediction.

Fig. 2 and Fig. 3 show the comparison for different encoding methods in donor and acceptor sites prediction. With the exception of binary vector encoding approaches (including 4-bit and 16-bit), the performances of MCM and WAM encoding methods are the worst ones, and the performances of MCM with DTF, MCM with UTF and WAM with DTF are

[2] LIBSVM is available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

the best ones. That is, SVM can benefit largely from the information contained in the false splice sites. According to [4], the performance of MCM encoding method is better than that of WAM encoding method, because WAM only takes into account the observed frequencies of pair of nucleotides and do not necessarily model the dependencies between nucleotides, which is again justified in acceptor sites prediction. However, it is not so for donor sites prediction. Maybe the difference is related to the optimal context lengths around the candidate splice site, i.e., $l_1$ and $l_2$ (see above), which are not completely identical for each encoding method [2]-[3]. The performance of MCM with UTF is better than that of WAM with UTF nearly in all cases. The performance of MCM with DTF matches that of WAM with DTF in all cases. In some case, the performance of MCM with UTF is better than those of MCM with DTF and WAM with DTF.

Generally speaking, it will greatly contribute to the prediction performance to take explicitly into account of the dependences among adjacent positions in the splice sites. However, for 4-bit binary vector encoding method, its performance is surprisingly good, which is possibly one of the reasons why it is commonly used. Furthermore, 16-bit binary vector encoding method does not show an obvious advantage. Maybe the number of the training data is inadequate relative to the dimension of input space, especially for the acceptor sites prediction (5788 vs. 1440).

# 8. Conclusions

From the above comparison of the performances, it is not difficult to see that the false splice sites can help improve the splice sites identification, which is again justified. Whether for donor sites prediction or for acceptor sites prediction, MCM with DTF and WAM with DTF should be good but not the best choices. It is still necessary to study other ways that make use of the information contained in the false splice sites as well as in the true splice sites. However, since the computational complexity of binary vector encoding methods is much smaller than that of others, and their performance is surprisingly good, their potentials need to be further investigated. In addition, it is also possible that the performances can be improved by combining more properties of sequences that affect the mechanism of RNA-splicing, e.g., compositional information, coding potential, etc.

# Availability

The source codes used in implementing the present methods are available from the authors upon request for academic use.

# Acknowledgement

This work is partially supported by the National Natural Science Foundation of China (Grant No. 60673122).

# References

[1] M. Yin and T.L. Wang, Effective hidden Markov models for detecting splicing junction sites in DNA sequences, *Information Sciences*, 139: 139-163, 2001.

[2] J. Huang, T. Li, K. Chen and J. Wu, An approach of encoding for prediction of splice sites using SVM, *Biochimie*, 88: 923-929, 2006.

[3] S. Degroeve, Y. Saeys, B.D. Baets, P. Rouzé and Y.V. Peer, SpliceMachine: Predicting splice sites from high-dimensional local context representations, *Bioinformatics*, 21: 1332-1338, 2005.

[4] A.K.M.A. Baten, B.C.H. Chang, S.K. Halgamuge and J. Li, Splice site identification using probabilistic parameters and SVM classification, *BMC Bioinformatics*, Suppl. 5(S15), 2006.

[5] S. Sonnenburg, Newmethods for splice site recognition. Diploma Thesis, Australian National University, 2002.

[6] M. Yamamura and O. Gotoh, Detection of the splicing sites with kernel method approaches dealing with nucleotide doublets, *Genome Informatics*, 14: 426-427, 2003.

[7] Y.F. Sun, X.D. Fan and Y.D. Li, Identifying splicing sites in eukaryotic RNA: support vector machine approach, *Computers in Biology and Medicine*, 33: 17-29, 2003.

[8] H. Ogura, H. Agata, M. Xie, T. Odaka and H. Furutani, A study of learning splice sites of DNA sequence by neural networks, *Computation Biology Medicine*, 27: 67-75, 1997.

[9] S.L. Salzberg, A.L. Delcher, S. Kasif and O. White, Microbial gene identification using interpolated Markov models, *Nucleic Acids Research*, 26: 544-548, 1998.

[10] S.L. Salzberg, M. Pertea, A.L. Delcher, M.J. Gardner and H. Tettelin, Interpolated Markov models for eukaryotic gene finding, *Genomics*, 59: 24-31, 1999.

[11] V.N. Vapnik, *The nature of statistical learning theory*, 2nd Edition, Springer Verlag, New York, 1999.

[12] V.N. Vapnik, *Statistical learning theory*, Wiley, New York, 1998.
[13] D. Kulp, D. Haussler, M.G. Reese and F.H. Eeckman, A generalized hidden Markov model for the recognition of human genes in DNA, *ISMB-96*, pp. 134-142, 1996.
[14] M.G. Reese, F.H. Eeckman, D. Kulp and D. Haussler, Improved splice site detection in Genie, *Journal of Computational Biology*, 4: 311-323, 1997.
[15] C. Burge and S. Karlin, Prediction of complete gene structures in human genomic DNA, *Journal of Molecular Biology*, 268: 78-94, 1997.
[16] R. Durbin, S. Eddy, A. Krogh and G. Mitchison, *Biological sequence analysis: Probabilistic models of proteins and nucleic acids,* Cambridge University Press, Cambridge, 1998.
[17] L.S. Ho and J.C. Rajapakse, Splice Site Detection with a higher-order Markov model implemented on a neural network, *Genome Informatics*, 14: 64-72, 2003.
[18] C.W. Hsu, C.C. Chang and C.J. Lin, A practical guide to support vector classification, http://www.csie.ntu.edu.tw/~cjlin/libsvm/.
[19] C. Mathé, M.-F. Sagot, T. Schiex and P. Rouzé, Current methods of gene prediction, their strengths and weaknesses, *Nucleic Acids Research*, 30: 4103-4417, 2002.
[20] U. Ohler and M.G. Reese, Detection of eukaryotic promoter regions using stochastic language models, *Molekulare Bioinformatik*, 89-100, 1998.
[21] U. Ohler, S. Harbeck, H. Niemann, E. Nöth and M.G. Reese, Interpolated Markov chains for eukaryotic promoter recognition, *Bioinformatics*, 15: 362-369, 1999.
[22] R. Staden, Computer methods to locate signals in nucleic acid sequences, *Nucleic Acids Research*, 12: 505-519, 1984.
[23] T.A. Thanaraj, Positional characterisation of false positives from computational prediction of human splice sites, *Nucleic Acids Research*, 28: 744-754, 2000.
[24] V. Vacic, L.M. Iakoucheva and P. Radivojac, Two sample logo: A graphical representation of the differences between two sets of sequence alignments, *Bioinformatcis*, 22: 1536-1537, 2006.
[25] G. Yeo and C.B. Buge, Maximum entropy modeling of short motifs with applications to RNA splicing signals, *Journal of Computational Biology*, 11: 377-394, 2004.
[26] M.O. Zhang and T.G. Marr, A weight array method for splicing signal analysis, *Computer Applications in the Biosciences*, 9: 499-509, 1993.