COMPUTATIONAL METHODS FOR SPLICE SITE PREDICTION

A dissertation
SUBMITTED TO THE FACULTY OF TECHNOLOGY
AND THE GRADUATE SCHOOL IN BIOINFORMATICS AND GENOME RESEARCH
of the University of Bielefeld
in partial fulfilment of the requirements
for the degree of
Doctor rerum naturalium (Dr. rer. nat.)

Leila Taher
March 2006

Adviser: Prof. Jens Stoye
Co-adviser: Prof. Burkhard Morgenstern

Abstract

Completing the genome sequence of a given organism is just the beginning of a series of subsequent tasks, namely, the discovery of the corresponding sequence complexities. For eukaryotic organisms, this refers especially to the documentation of the coding exons of each gene, as well as non-coding and regulatory sequences. In spite of the extensive research done in the area, the production of genomic data remains far ahead of the ability to reliably predict such features computationally.

Eukaryotic genes consist predominantly of exons and introns; while the introns are non-coding regions that are removed from the primary transcript during RNA splicing, the exons are the coding regions that are spliced together during the mRNA maturation process. The borders between exons and introns present conserved dinucleotides, and are called splice sites; the intron-exon boundary is referred to as *acceptor splice site* and the exon-intron boundary as *donor splice site*.

Accurate eukaryotic gene prediction clearly requires exact splice site prediction methods. There are different pattern recognition techniques available to assess the quality of candidate splice sites; most of them proceed by computing a score derived from models about the distribution of the nucleotides in the neighbourhood of a splice site consensus sequence. Using Support Vector Machines (SVM) and a kernel particularly designed for the study of biological sequences, we investigate which pattern occurrences in which positions are relevant for splice site prediction. In addition, we propose a splice site prediction method that improves traditional position-specific weight matrices. Finally, we contribute with an intron model that relies on secondary structure information and demonstrates the ability to distinguish intron sequences from random data.

Acknowledgements

I would firstly like to thank my supervisors, Prof. Jens Stoye nad Prof. Burkhard Morgenstern, for their guidance during this thesis. I would also like to thank Dr. Peter Meinicke for additional support and advice, as well as many useful discussions.

I greatly appreciate the support team at the Bielefeld University Bioinformatics Server, especially Alexander Sczyrba and Henning Mersch, for their help in setting up the web service for AGenDA.

I also thank the staff of the NRW International Graduate School in Bioinformatics and Genome Research, for giving me the opportunity to study at the University of Bielefeld, and all my friends and colleagues from the graduate school for making my stay in Germany both scientifically and socially rewarding.

Last but not least, I specially thank my entire family and friends, for their love and continuous encouragement during the time that I was engaged in this study.

This work was funded by the Nordrhein-Westfalen Government (Germany).

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A major objective of any genome sequencing project is the identification of all genes, together with the corresponding proteins, their regulation and functions. In view of the foregoing, the prediction of genes has become one of the most important problems in computational biology.

Most attempts to solve this problem are based on statistical characteristics of known genes. Recent advances in large-scale genomic DNA sequencing have opened a way toward a new approach to gene recognition that uses genomic sequences from related organisms as a clue for recognising new genes.

As evolution progresses and mutations randomly occur in the genome, some mutations are more likely to survive. Specifically, mutations which cause frame shifts in the coding regions of important proteins will most likely not survive. By contrast, mutations in non-functional regions will have very little effect on the survival of the species. So, the proposition is that orthologous protein coding regions in related organisms will be more similar than orthologous non-gene regions. The converse proposition is that very similar orthologous regions will be most likely gene-regions, instead of non-gene regions. Thus, an analysis of the highly conserved orthologous regions in two different genomes will certainly yield relevant information about the gene coding regions which are common to both.

However, despite the attention received in the last years, there is still place for improvement [Guigó et al., 2000].

The coding regions of most eukaryotic genes (*exons*) are interrupted by non-coding (*introns*) regions. The border going from exons to introns is called *donor site* (or 5' boundary), and the border separating introns from exons is called *acceptor site* (or 3' boundary). Donor sites are characterised by the occurrence of the consensus dinucleotide GU (GT in the DNA sequence) and acceptor sites by the consensus dinucleotide AG (in the 5' to 3' direction); however, the presence of such conserved dinucleotides does not constitute a sufficient condition for splicing. As it can be deduced from the above mentioned, the accurate prediction of splice sites is an issue of critical importance for eukaryotic gene prediction, for which solutions remain limited [Mathé et al., 2002].

1.1 Contribution of this thesis

This thesis presents and evaluates three novel approaches to the prediction of constitutive splice sites, with different degrees of success.

The first approach is based on *Support Vector Machines* (SVMs). A SVM is a supervised classification method that deduces a decision function from a training data set, which can be in turn used to classify new data. The input for our Support Vector Machines is a representation of the nucleotide sequences in terms of the substrings of lengths 1, 2, 3, 4, 5, and 6 that they contain, associated with a variable degree of information about their location. The proposed sequence representation has the advantage of allowing a thorough study of the prediction of splice sites using SVMs. We show how the discrimination function of the model can be directly used to gain insight into the sequence patterns that are relevant for the splicing mechanism, and thus relate the performance of the method to sequence properties. This analysis exemplifies how such a learning system can be useful not only as a prediction tool, but also as a method to identify characteristics that might have a biological meaning. To conclude, we compare our findings with the output of standard feature selection methods, techniques developed to reduce the input to a set of patterns such that the classification system produces the optimal performance.

The second contribution of this thesis regarding splice site prediction is a heuristic method to improve the performance of *position-specific weight matrices* (WMMs). Position-specific weight matrices and Markov chains are the method of choice for splice site prediction (see Cai et al. [2000]). Higher-order models perform normally better than lower-order models, but they frequently have a number of limitations associated with high state-space complexity and reduced coverage, resulting sometimes in an even worse overall prediction accuracy. We show how Gaussian smoothing can help overcome this issue, and propose a model that proves to perform very well on the splice site prediction problem, and could be integrated in the future to gene prediction programs. Gaussian smoothing is a filter commonly applied in order to reduce noise in image processing; hereby, we demonstrate how the properties of Gaussian functions can also be useful to integrate information about the presence or absence of signals at neighbouring positions in biological sequences, and how that influences the performance of a splice site prediction system.

Finally, we evaluated the prediction of splice sites using models that describe secondary-structure properties of the intron sequences. Functional RNA secondary structure is frequently conserved during evolution, which implies that pairing bases covary; that is, if a mutation occurs on a nucleotide such that the secondary-structure is disturbed, then there will be a compensatory mutation on the corresponding pairing nucleotide [Hofacker et al., 2002]; models capturing such covariations between nucleotides are called *covariation models*. Given that the mechanism of splicing implies interactions between different parts of the pre-mRNA molecule, we assumed that these could be associated with specific spacial requirements. Based on this information, we created a multiple alignment of intron sequences and secondary-structure consensus, using available software [Eddy and Durbin, 1994]. Although our model is not optimal for distinguishing real intron sequences from overlapping pseudo intron sequences, it is successful in a more general setup.

We also present a web server for the gene prediction program AGenDA [Rinner and Morgenstern, 2002]. AGenDA is a gene prediction program that compares the target sequence with an evolutionary related sequence from another species in order to find probable orthologous regions, and then combines this information with basic and general statistical models. This is intended to make the method as universal as possible, and therefore, applicable to different species without any major changes and with an equivalent performance. The web server automatises the processes that create the input for the program. We also improved the algorithm that retrieves the optimal gene structure from the output list of exons produced by AGenDA [Taher et al., 2003, 2004b].

1.2 Protein synthesis in eukaryotic organisms

Figure 1.1 summarises the steps and molecules involved in eukaryotic protein synthesis. Initially, the genomic DNA is used as template for the synthesis of an intermediate molecule, messenger RNA (mRNA), in a process known as *transcription*, which occurs within the cell nucleus and is facilitated by an RNA polymerase and transcription factors. The mRNA is subsequently exported to the cell cytoplasm, where the information for the amino acid sequence it encodes (originally encoded by the DNA) is retrieved and converted into a protein on the ribosomes, during the *translation*. Transfer RNAs (tRNAs), specific to each amino acid, transport amino acids to complementary binding sites on the mRNA bound to the ribosomes.

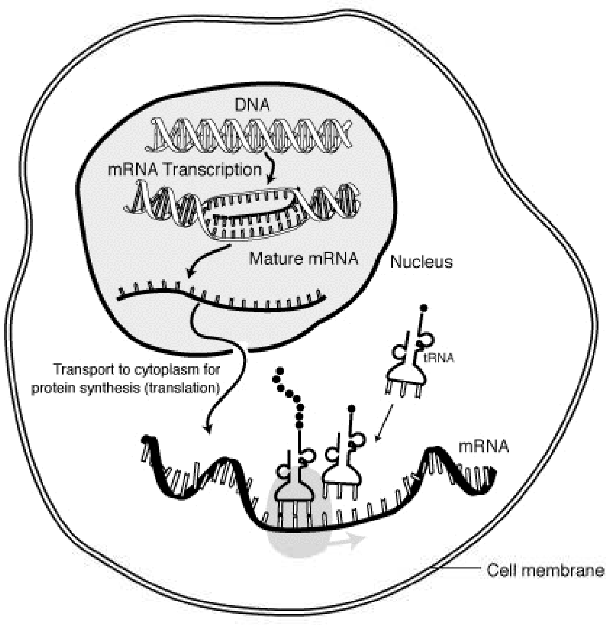


Figure 1.1: Protein synthesis or genetic information flow in an eukayrotic cell.

1.2.1 mRNA post-transcriptional processing

The primary transcription product of eukaryotes is called heterogeneous nuclear RNA (hnRNA). This mRNA precursor (pre-mRNA), whose structure is shown in Figure 1.2, goes through a series of modifications [Alberts et al., 2002] that are required for its transport to the cytoplasm, or/and to improve the stability of the molecule or its interaction with the ribosomes. The completely processed mRNA precursor is termed mature mRNA, or simply mRNA, and its structure is schematised in Figure 1.3. Post-transcriptional modifications comprise:

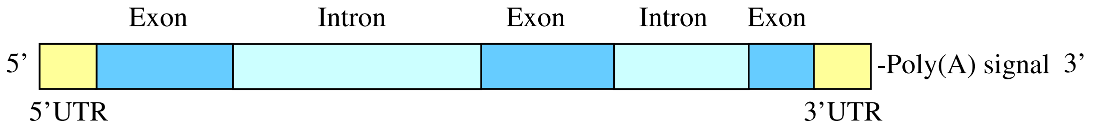


Figure 1.2: Primary RNA transcript.

- 5' cap addition: 7-methyl guanosine is added to the 5' end of the pre-mRNA. This modification is critical for the recognition and proper attachment of mRNA to the ribosome, and may also be important for processes such as splicing and transport.

- Splicing: Pre-mRNA sequences include two different types of segments: *exons* and *introns*. Exons often encode for protein, while introns do not and must be excised before translation, by a process called splicing.

- Polyadenylation: Polyadenylation is the enzymatic cleavage at the 3' end of the RNA molecule, and the subsequent addition of 80 to 250 adenosine residues to the free 3' end at the cleavage site. This poly(A) tail protects mRNA from premature degradation by exonucleases.

- Editing: Occasionally, a pre-mRNA will be edited, changing the nucleotide composition of the corresponding mRNA; examples of editing include tissue-specific edition to create early stop codons, and thereby, shorter proteins.

The coding regions of mRNA are composed of codons, which are translated into amino acids, the building blocks of proteins, by the ribosomes. Coding regions begin with the start codon (AUG) and end with one of three possible stop codons (UAA, UAG, or UGA). Additionally, some coding regions may not code for protein, but serve as regulatory sequences. The coding regions of mRNA are preceded and succeeded by untranslated regions (5' UTR and 3' UTR, respectively), which stabilise mRNA molecules and improve translation efficiency.

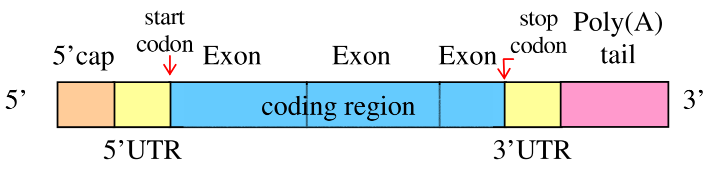


Figure 1.3: Mature mRNA structure.

In the context of gene prediction, the most relevant post-transcriptional modification is splicing, because it delimits the sequence of the protein encoded by a gene. Indeed, the fact that genomic sequences, the main input for gene prediction programs, contain variable-length introns makes eukaryotic gene prediction challenging when compared to prokaryotic gene prediction.

1.3 Splicing

The term splicing refers to the post-transcriptional modification of RNA that results in the removal of introns. This process is not restricted to the nucleus, but also occurs in organelles; however, it proceeds through a different mechanism. Indeed, besides the introns found in the nucleus, or *nuclear introns*, there have been described two other types:

- Group I introns, which are found in organelles and bacterial genes along with some lower eukaryotes nuclear genes, and whose splicing does not require the aid of proteins (this is known as *self-splicing*), but the presence of GTP.

- Group II introns, which are found in organelles and bacterial genes, able to self splice without the aid of proteins; their sequence and mechanism distinguishes them from Group I.

This thesis addresses the problem of eukaryotic gene prediction, and therefore focuses on splicing of introns in nuclear genes, whose mechanism requires proteins and other RNAs.

1.3.1 Splicing consensus sequences

Most of the nucleotide sequence of introns can be modified without causing a significant alteration of gene function, and the only highly conserved patterns they contain are those required for splicing. Most introns start with the dinucleotide GU (GT in the original DNA sequence) and end with the dinucleotide AG (in the direction 5' to 3'), and these signals are referred to as *donor sites* and *acceptor sites*, respectively. More generally, intron borders are called *splice sites*.

The occurrence of the aforementioned dinucleotides upstream and downstream is not sufficient to signal the presence of an intron. Another distinctly important conserved pattern is the branch site, with consensus sequence *(C/T)N(C/T)(C/T)(A/G)**A**(C/T)*, where **A** is conserved in all genes, and located 20 - 50 bases upstream of the acceptor site [Rautmann and Breathnach, 1985]. There is also evidence for a pyrimidine-rich region preceding acceptor sites [Brow, 2002].

Splice site consensus regions for mammalians [Jacob and Gallinaro, 1989, Strachan and Read, 1999] are depicted in Figure 1.4. The symbols correspond to the nomenclature for nucleic acid sequences proposed by IUPAC, i.e., M corresponds to A or C, Y to pyrimidines (T,U or C), and R to purines (A or G).

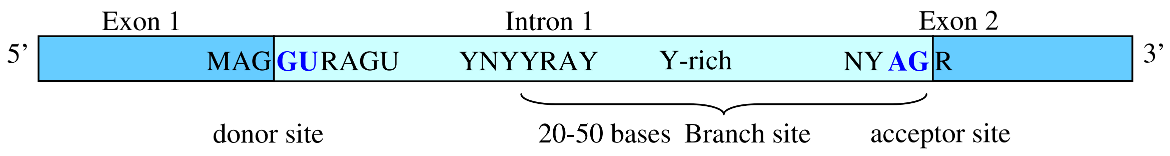


Figure 1.4: Splice site consensus regions.

1.3.2 Exon/Intron evolution

Ever since introns were discovered [Berget et al., 1977], there has been debate going on in relation to their evolutionary origin and significance. This has led to the formulation of two main opposing theories, known as *introns-early* theory and *introns-late* theory.

Originally Gilbert [1978] suggested that introns would allow exon shuffling, which would facilitate the increase of genetic complexity by differential splicing and accelerate evolution by permitting new rearrangements via recombination to construct new genes (and hence, proteins).

Subsequent to this, several authors [Gilbert, 1987, Blake, 1978] suggested that this theory implies that each exon would code for a unit of protein that has some integrity by itself. Proteins are indeed composed by domains or subunits, such as helices and beta strands, but it has been shown that exons do not always map to those domains [Stoltzfus et al., 1994]; this fact is especially detrimental to the introns-early theory.

Now, if the former assumptions are true, it would be highly unlikely that introns would have evolved late to interrupt genes; it seems more probable that nuclear introns would have been present in the common ancestor of all presently living organisms; bacterial genomes would have originally contained introns, but were afterwards streamlined for competitive growth advantage.

All of these speculations were reunited in the *introns-early* theory, often also referred to as the *exon theory of genes.*

The opposing *introns-late* hypothesis [Logsdon et al., 1995, Cavalier-Smith, 1985] proposes that ancient gene structures contained no introns, and that these arose and evolved only in eukaryotic genomes. This theory suggests that introns were randomly inserted into previously continuous genes that correspond to ancestral forms, and were similar to the current prokaryotic genes [Cavalier-Smith, 1991]. It has been speculated that introns could have been originated by events such as gene duplication, transposable elements, or self insertion. However, the introns-late theory has difficulties explaining the function of introns.

The gene encoding the enzyme triose-phosphate isomerase (TPI; EC 5.3.1.1) has been fundamental to the controversy, as it presents evidence supporting both theories. On the one hand, correlation analyses between intron positions and TPI protein structure hint that the gene was indeed assembled by exon shuffling. Additionally, some TPI introns have been found to be conserved between animals and plants, and statistical calculations indicate that ancient introns were once phase 0 introns, i.e., introns that do not interrupt a codon [Roy, 2003, de Souza et al., 1998]. All these findings support the introns-early theory. On the other hand, TPI gene sequences from other species present introns that disrupt previously recognised gene/protein structure correlations, in a consisting manner with a random model of intron insertion. As each of these introns is mainly present in a single species, they seem to be recent. These results, presented by Logsdon et al. [1995], support the introns-late theory.

Finally, it is also argued that both opposing theories could be reconciled; some introns may have been early while others may have been inserted later [Tyshenko and Walker, 1997].

1.3.3 Splicing mechanisms

The mechanism of splicing is complex [Green, 1991]. Introns are removed in a two-step enzymatic reaction that involves the formation of a large multicomponent ribonucleoprotein catalytic complex, the *spliceosome*. The spliceosome is assembled from the small nuclear ribonucleoproteins (snRNPs) U1, U2, U5 and U4/U6, and other components.

The first step is a hydrophilic attack. The 2' hydroxyl group of the conserved adenosine within the branch site attacks the conserved guanosine of the donor splice site at the junction between the exon and the intron.

In the first step of the reaction, the 2' hydroxyl group of the conserved A nucleotide within the branch site attacks the phosphate group in the conserved G nucleotide of the donor splice site and cleaves it; thereby, the 5' end of the intron sequences becomes covalently linked to this A nucleotide. This step is sketched in Figures 1.5 and 1.6.

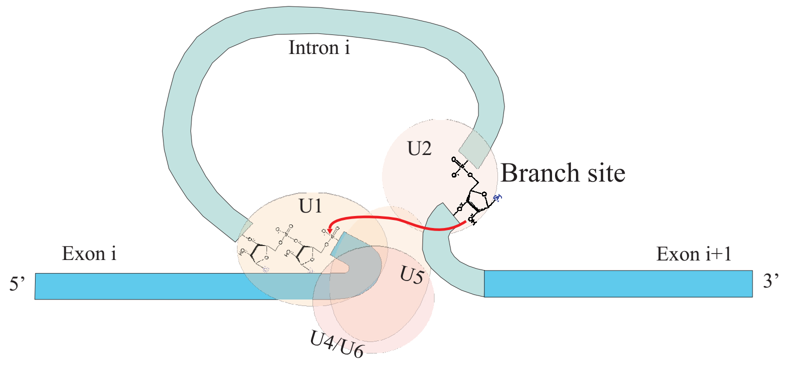


Figure 1.5: First step of the splicing catalytic reaction (a): nucleophilic attack by the 2' hydroxyl group.

In the second step (see Figures 1.7 and 1.8), the 3' hydroxyl group of the last nucleotide in the released exon (*exon i*) attacks the phosphate group connecting the intron to the next exon (*exon i+1*), cleaving the RNA molecule at the acceptor splice site; the two exon sequences join and the intron sequence is released in the form of a 2'-5'-phosphodiester RNA lariat, which will be then degraded in the nucleus.

Splice sites operate in generic pairs. Therefore, if the end of an intron (acceptor site) is mutated, that intron plus the following exon and next intron will be spliced out (see, for example, Vandenbroucke et al. [2002]).

Sometimes pre-mRNA may be spliced in several different ways, allowing a single gene to encode two or more proteins. This process is called *alternative splicing*, and has been observed in more than 60% genes in the human genome [Croft et al., 2000]. Nevertheless, the number of exons which are

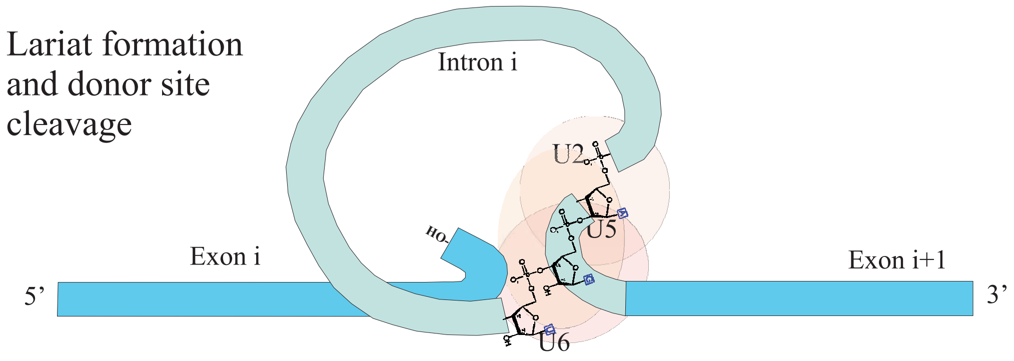


Figure 1.6: First step of the splicing catalytic reaction (b): lariat formation and donor site cleavage.

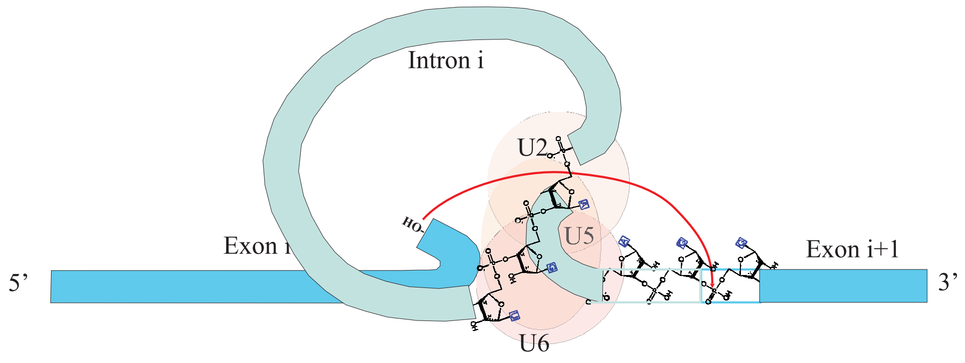


Figure 1.7: Second step of the splicing catalytic reaction (a): nucleophilic attack by the 3' hydroxyl end of the released exon.

subject to alternative splicing in a gene is very low, and, in contrast, the vast majority of the exons is *constitutively* spliced.

Although, some gene transcripts have been shown to loose their introns in a consistent order, the current model considers that the hnRNA adopts different conformations after specific introns are removed, subsequently making other introns available for removal. As a conclusion, the removal of introns does not proceed sequentially along the transcript.

The role of small nuclear ribonucleoproteins

There exist five major small nuclear ribonucleoprotein particles (snRNPs), called U1, U2, U4, U5, and U6, which form the core of the spliceosome. In addition, numerous other snRNPs are present in the nucleus in smaller numbers. The structure of most snRNPs consists of one small nuclear RNA molecule (snRNA), ranging in length from 107 to 210 nucleotides, complexed with six to ten proteins [Lodish et al., 2000].

The assembly of the spliceosome begins with the recruitment of U1 snRNP to the donor splice site. The donor site consensus sequence is complementary to the 5' end of U1 snRNP; additionally, extrinsic factors contribute to the stability of the interaction. These extrinsic factors include the members of the SR protein family, a group of highly conserved RNA-binding proteins characterised by at least one N-proximal RNA recognition motif (RRM) and a C-proximal domain rich in serine and arginine (an RS domain), involved in protein-protein interactions [Hastings and Krainer, 2001].

The next step in the formation of the spliceosome is the specific binding of U2 snRNP to the branch site. Although this interaction is based on complementary pairing of conserved residues,

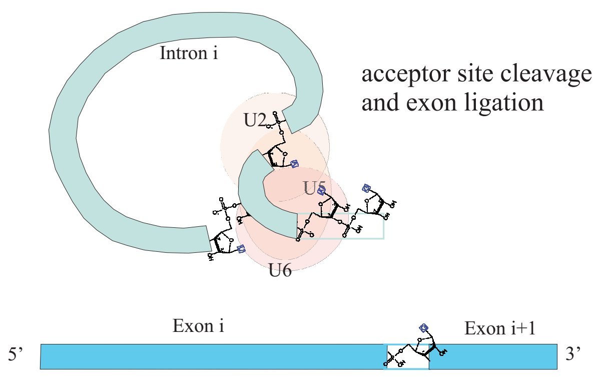


Figure 1.8: Second step of the splicing catalytic reaction (b): excision of the intron sequence and ligation of the exon sequences.

requires the extrinsic factors BBP (branch-point binding protein) and U2AF (U2 snRNP auxiliary factor), which bind specifically to the poly-pyrimidine tract upstream adjacent to the acceptor splice sites and positions the U2 snRNP on the branch site. Human U2AF is a heterodimer; the large subunit contains three RNA recognition motifs (RRM), which are responsible for the binding to the poly-pyrimidine tract, and an N-proximal domain rich in serine and arginine (RS domain), which would facilitate the binding of U2 snRNP to the branch site, probably by shielding the negative charge of the RNA phosphodiester backbone; the small subunit contains an RS domain that would mediate interactions with SR proteins that stabilise binding of U2AF to the RNA molecule and the spliceosome complex[Zamore et al., 1992].

U4, U5, and U6 snRNPs interact to form the U4-U5-U6 tri-snRNP complex. U4 and U6 snRNPs are held together by a hydrogen-bonding; U4 snRNP attracts U6 to the spliceosome and prevents unfavourable associations. After binding of the U1 and U2 SnRNPs to the pre-mRNA, U5 snRNP binds to a conserved exon sequence adjacent to the donor site, and the tri-snRNP complex joins in to complete the formation of the spliceosome [Turner et al., 2004].

After the first step of the catalytic reaction, the interaction between U1 snRNP and the intron sequence is destabilised and U6 snRNP replaces U1 snRNP. This action would disrupt the bond between U6 and U4, allowing U6 to bind to both the donor site and the branch point. Finally, during the second step of the catalytic reaction, U5 snRNP establishes an additional interaction with a sequence immediately downstream of the acceptor splice site, aligning the 5' exon with the 3' exon after the first step of the reaction [Alberts et al., 2002].

The recognition of a particular splice site depends on the strength of the consensus sequence, but also on the interaction of specific nucleotide sequences in the pre-mRNA (*cis-acting elements*) with numerous SR proteins, hnRNPs and snRNAs (*trans-acting factors*) that regulate the tissue-specific constitutive and alternative splicing of different exons. [Lodish et al., 2000].

Chapter 2

Methods

2.1 Learning systems and binary classifiers

A classifier is a function that assigns an instance represented by attributes to a class. In the case of a binary classifier, the instances or examples belong to either of two classes, commonly labelled as -1 and $+1$. Let X be the set of attributes of the instances, a binary classifier is characterised by the function

$$f : X \to \{-1, +1\}. \tag{2.1}$$

In medicine, binary classifiers are commonly employed to discriminate between test results; either the sample belongs to a normal person or to a person who has the disease the test was designed for. In this case, the *instances* or *examples* are the samples taken from the patients, and the *attributes* or *features* are the measured parameters.

The prediction of splice sites can be clearly seen as a classification problem. Basically, given the occurrence of the dinucleotide AG or GT in a particular context in a RNA sequence, we are interested in deducing a rule to distinguish between the dinucleotides that indicate real splice sites (*positive examples*) from those which do not (*negative examples*).

2.2 Experiment design and evaluation datasets

The performance of a learning system depends on the data, and a good estimation of its performance requires systematic train-and-test experiments [Pyle, 2001]. The number of repetitions necessary depends on the size of the sample; for small samples, multiple train-and-test experiments are normally preferred.

The classification experiments presented here were done using mainly two data partitioning schemes, i.e., cross-validation and random subsampling.

2.2.1 Random subsampling

In this scheme, the data is independently and randomly separated into two subsets, one for training and the second one for testing. This process is repeated a sufficient number of times, in order to obtain results that can be judged by a significance test. Each repetition of the train-and-test experiments produces a new classifier. The estimated error rate is the average of the error rates of all the classifiers produced. Random subsampling normally results in better error estimates than a single train-and-test partition.

2.2.2 k-fold cross-validation

Cross-validation is technically a especial case of random subsampling, in which the data is divided into several subsets. In this scheme, the n instances in the dataset are partitioned into k subsets; a classifier is then trained using $n \cdot \frac{k-1}{k}$ instances and tested on the remaining $\frac{n}{k}$ instances. This method is computationally expensive, but generally performs well.

The most common scheme employed is a 3-fold 5-fold cross-validation scheme. The data is initially divided into three subsets, two for training the classifier and one for testing its performance. The two training subsets are subsequently divided into five subsets, from which four are used for training and one for testing. In the internal loop takes place the parameter selection; the external loop evaluates the classifier with optimal parameters. Figure 2.1 illustrates this scheme.

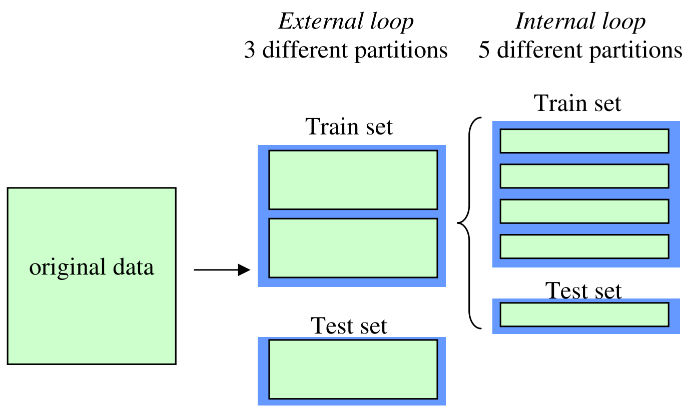


Figure 2.1: 3-fold 5-fold cross-validation scheme.

2.2.3 Datasets

The tests presented in this thesis were performed on two sequence datasets.

For the experiments about splice site prediction, we employed the HS3D (*Homo sapiens* Splice Sites Dataset, [Pollastro and Rampone, 2002, 2003]), a dataset of *Homo sapiens* exon, intron and splice regions extracted from GenBank Rel.123. This dataset was constructed with 697 entries of human nuclear DNA including genes with complete CDS and with more than one exon extracted from the GenBank Release 123 (8436 entries for primate sequences); from these entries were extracted 4450 exons and 3752 introns, and in turn 3762 donor and 3762 acceptor sites as windows of 140 nucleotides around each splice site. Finally, sequences not including canonical junctions (i.e., containing the conserved dinucleotides GT and AG), sequences with insufficient data for a 140 nucleotide window, and sequences including characters different from A, C, G or T were discarded. The final dataset contains 2796 donor sites and 2880 acceptor sites. In addition, there are 271,937 windows of false donor sites, and 332,296 windows of false acceptor splice sites, which were selected by searching GT and AG dinucleotides in non-splicing positions; there are 21,222 false acceptor sites and 17,431 false donor sites in a range of ± 60 from true splice sites, which are marked as *proximal* splice sites.

In addition, we used a set of 117 of orthologous sequences from human and mouse compiled by Batzoglou et al. [2000]. These sequences were extracted from a dataset of 1196 corresponding human and mouse mRNA pairs assembled by Makalowski et al. by identifying orthologous pairs of genes between human and murine sequences from GenBank Release 87; this compilation involved a detailed phylogenetic study of the sequences. Batzoglou et al. [2000] discarded the sequences containing genes in the human or mouse that did not have a corresponding entry in the other organism, as well as those containing only part of the exons or/and introns corresponding to a particular gene. The final 117 sequences are annotated with respect to gene structure.

2.3 Classification methods

The most commonly classification methods applied to the splice site prediction problem are Support Vector Machines, Markov chain classifiers, position-specific weight matrices, maximal dependence decomposition models, probabilistic or Bayesian networks, and Naïve classifiers. This section describes the theoretical background for each of them.

2.3.1 Support Vector Machines

Support Vector Machines (SVMs, [Cristianini and Shawe-Taylor, 2000]) are learning systems that infer classification or regression functions from a set of training data using a learning algorithm from optimisation theory, whose criteria are derived from statistical learning theory [Vapnik, 1998].

A general classification problem can be formulated as follows:

Let us consider two random variables X and Y with possible values in $\Omega_X \subseteq \mathbb{R}^n$ and $\Omega_Y \subseteq \mathbb{R}$, respectively, related by a probabilistic relationship $P(X, Y)$ defined over the set $\Omega_X \times \Omega_Y$. Given a training set S obtained by sampling m times the set $\Omega_X \times \Omega_Y$ according to $P(X, Y)$, that is, a set S of observations $x_1, x_2, \ldots, x_m \in \Omega_X \subseteq \mathbb{R}^n$ with associated labels $y_1, y_2, \ldots, y_m \in \Omega_Y \subseteq \mathbb{R}$,

$$S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\},$$

the objective is to provide an estimator h of the function f

$$f : \Omega_X \to \Omega_Y, \tag{2.2}$$

which can not only predict the labels of the training data, but also the label $y \in \Omega_Y$ of any unseen observation $x \in \Omega_X$. This function h is further referred to as *classification* or *decision function.*

Usually, the labels only take values in $\{-1, +1\}$, and the label of an observation x can be realised by calculating the sign of the values of $f(x)$ and $h(x)$.

We apply SVMs to the classification or prediction of splice sites. In our case, the observations $x_1, x_2, .., x_m$ are the candidate splice site sequences, i.e., nucleotide sequences containing either the dinucleotides AG (acceptor splice sites) or GT (donor splice sites), in an *appropriate* mathematical representation. Labels $y_1, y_2, .., y_m$ classify the signals either as *real (true) splice sites* or *pseudo (false) splice sites.* To be consistent with common nomenclature, we identify real splice sites with the label $+1$, and pseudo splice sites with the label -1. The goal is to use the available observations (*training data*) to deduce a *classification function* f that is able to generalise well, that is, which is also able to correctly classify further splice site candidates x (also referred to as testing data) as *true* or *false.*

For practical reasons, it might be necessary to restrict the space the *classification function* h belongs to. Assume this restricted space is called H. As H is the space of the hypotheses that are available to approximate f, its choice might require prior knowledge of the problem. In addition, not every function h which is able to classify the training data well, will necessarily perform well on independent testing data.

Technically formulated, minimising the training error (or *empirical risk*) for the training set

S, defined as

$$R_{emp}(h) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}|f(x_i) - y_i|, \tag{2.3}$$

does not guarantee a small test error (or *risk*) on examples distributed according to the unknown distribution $P(X, Y)$

$$R(h) = \int_{x\in\Omega_X, y\in\Omega_Y} |f(x) - y|\ P(x, y)dx\ dy. \tag{2.4}$$

The connection between the *empirical risk* and the *real risk* depends on the hypothesis space H. If the space is very restricted, i.e., if the choice for the hypothesis function h is very limited, the empirical risk of the best hypothesis is probably close to the real risk. On the other hand, if the space H easily allows finding a hypothesis function with low empirical risk, this does not guarantee a good generalisation. Instead, this can be achieved by choosing h from a set of functions H which has a *capacity* suitable for the amount of available training data. Depending on both the empirical risk and the capacity of the hypothesis space H, statistical learning theory [Cristianini and Shawe-Taylor, 2000] provides bounds for the real risk; the minimisation of these bounds leads to the theory known as *structural risk minimisation.*

Given a parameter $\eta \in \mathbb{R}$ chosen so that $0 \leq \eta \leq 1$ and m training examples, with probability $1 - \eta$ the risk is bounded by the following inequality

$$R(h) \leq R_{emp}(h) + \sqrt{\frac{p(ln\frac{2m}{p} + 1) - ln(\frac{\eta}{4})}{m}}, \tag{2.5}$$

where p is the Vapnik-Chervonenkis (VC) dimension. The VC-dimension of a hypothesis space is a combinatorial property of that space H, which measures its *capacity.* It is defined as the maximum number of instances or points that can be shattered by the functions belonging to the hypothesis space. A set containing p points is shattered by a function if each of the 2^p subsets of that set of points can be separated from the rest of the points by a function in the hypothesis space.

The process of training a Support Vector Machine is essentially an implementation of the structural risk minimisation principle.

Linear Support Vector Machines

In their basic form, SVMs *learn* linear classification functions. In this case, the data belonging to two different classes are assumed to be separable by a linear function.

A hypothesis space H whose capacity or VC-dimension can be easily computed is composed by the hyperplanes with coordinate form of equation

$$w \cdot x + b = 0, \tag{2.6}$$

where $w \in \mathbb{R}^n$ is a vector normal to the hyperplane and $b \in \mathbb{R}$ must satisfy $b = -w \cdot x_0$ for every point x_0 in the hyperplane.

Vapnik and Chervonenkis [1964] considered this hypothesis space and the classification function

$$h(x) = sign(w \cdot x + b) \tag{2.7}$$

to propose an algorithm, the *Generalised Portrait*, for constructing the function h from separable

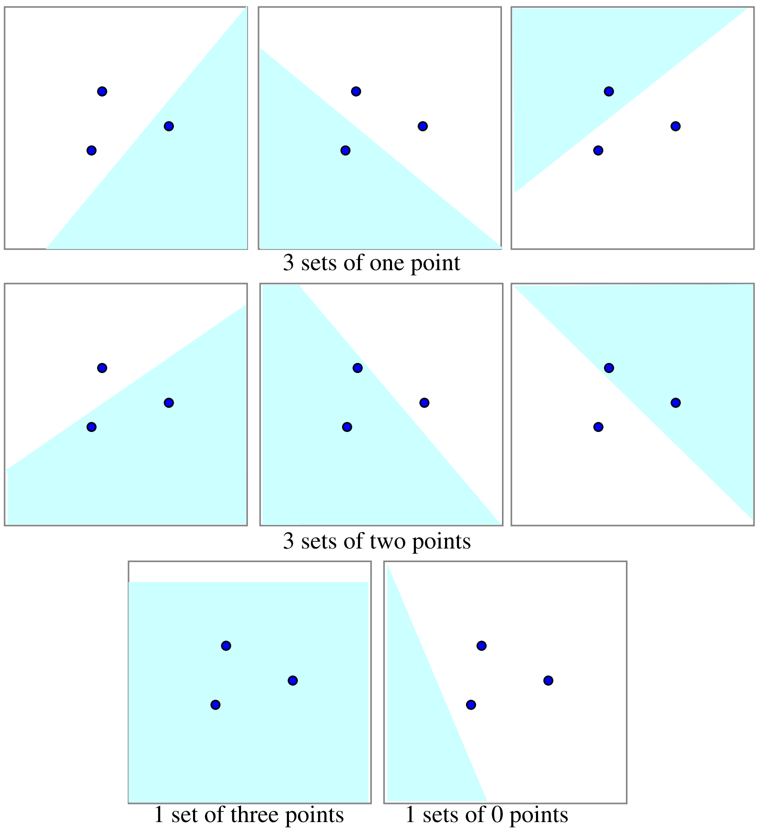


Figure 2.2: The VC-dimension of half-planes in $\mathbb{R}^2$ is 3. The graph shows the shattering of a set containing three points lying in a plain using half-plane functions.

training data.

Regarding the calculation of the VC-dimension of such a hypothesis space, let us study the VC-dimension of a set of half-planes in $\mathbb{R}^2$. It is clear that a set of three non-collinear points in $\mathbb{R}^2$ can be easily separated into all possible subsets of 0, 1, 2, or 3 points by a half-plane, which means that the VC-dimension is at least 3 (see Figure 2.2). In order to shatter four points, no three can be collinear. So, the four points must form the vertices of either a convex quadrilateral or a concave quadrilateral; in the first case, not all subsets with two points can be separated from the remaining elements, while in the second case, the vertex corresponding to the internal angle greater than 180^o cannot be isolated. Therefore, the VC-dimension of hyperplanes in $\mathbb{R}^2$ is 3.

The structural risk minimisation theory predicts that among all hyperplanes separating the training data, a solution to the classification problem resulting in the minimum risk is the hyperplane that has the *maximal margin* to the training set, that is, the hyperplane that maximises its distance to the nearest data points on each class. This linear classifier is termed the *optimal separating hyperplane*. Comparing the three different decision planes in Figure 2.3, the classifier with larger margin will have lower expected risk.

In order to separate two classes of training examples $x_1, x_2, \ldots, x_m \in \mathbb{R}^n$ with respective labels $y_1, y_2, \ldots, y_m \in \{-1, +1\}$, a decision hyperplane $w \cdot x + b = 0$ has to satisfy for every $i = 1, 2, \ldots, m$ the following equations

$$\begin{cases} w \cdot x_i + b > 0 & \text{if } y_i = 1 \\ w \cdot x_i + b < 0 & \text{if } y_i = -1 \ . \end{cases} \tag{2.8}$$

Among all the hyperplanes satisfying this condition, the *optimal separating hyperplane* is the hyperplane that separates the two classes with maximal margin.

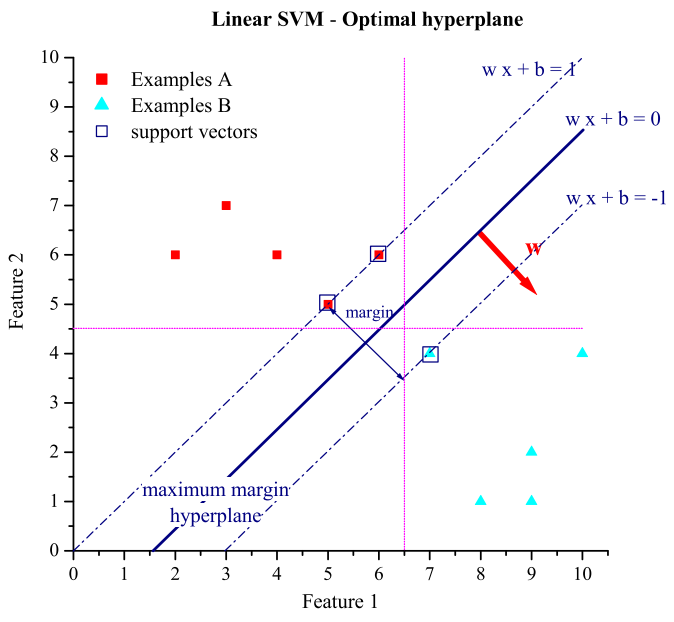


Figure 2.3: Optimal separating hyperplane. The graph shows 10 examples and three hyperplanes plotted into a coordinate system. The continuous line corresponds to the optimal hyperplane, while the dotted magenta lines represent non-optimal separating hyperplanes.

Without losing generality we can scale w and b by the same factor, and formulate an equivalent decision boundary as

$$\begin{cases} w \cdot x_i + b > 1 & \text{if } y_i = 1 \\ w \cdot x_i + b < 1 & \text{if } y_i = -1 \ , \end{cases} \tag{2.9}$$

or simply,

$$y_i(w \cdot x_i + b) \geq 1, \ i = 1, 2, \ldots, m. \tag{2.10}$$

The examples $x \in \mathbb{R}^n$ with label $y \in \{-1, +1\}$ that satisfy $y(w \cdot x + b) = 1$, i.e.

$$w \cdot x + b = 1 \quad \textit{or} \quad w \cdot x + b = -1, \tag{2.11}$$

are called *support vectors*. The vector w is called *weight vector*.

The distance of a point x_i to a hyperplane in $\mathbb{R}^n$ with equation $w \cdot x + b = 0$ is

$$d(w, b; x_i) = \frac{|w \cdot x + b|}{\|w\|}. \tag{2.12}$$

For $i = 1, 2, \ldots, m$, the margin $\rho(w, b)$ of a hyperplane is

$$\rho(w, b) = \min_{x_i : y_i = 1} d(w, b; x_i) + \min_{w_j : y_j = -1} d(w, b; x_j). \tag{2.13}$$

Substituting Equation 2.12 in Equation 2.13, the margin is given by

$$\rho(w, b) = \frac{2}{\|w\|}. \tag{2.14}$$

Therefore, finding the optimal hyperplane is equivalent to maximising the margin in Equation 2.14, subject to the constraints in Equation 2.10

$$\begin{array}{rc} \max & \rho(w,b) = \frac{2}{\|w\|} \\ \text{subject to:} & y_i(w \cdot x_i + b) \geq 1,\ i = 1, \ldots, m. \end{array} \tag{2.15}$$

The equation of the optimal hyperplane, defined in terms of the optimal solution of the problem, w^{opt}, is $w^{opt} \cdot x + b = 0$.

An equivalent, but perhaps more classical formulation for the problem presented in 2.15 is the following.

Given the set of linearly separable observations

$$S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m) \mid x_i \in \mathbb{R}^n \ \wedge \ y_i \in \mathbb{R},\ i = 1, 2, \ldots, m\},$$

the hyperplane that separates the data with maximal margin takes the form $w^{opt} \cdot x + b = 0$, where w^{opt} is the optimal solution of the optimisation problem given by

$$\begin{array}{rc} \min & \|w\|^2 \\ \text{subject to:} & y_i(w \cdot x_i + b) \geq 1,\ i = 1, \ldots, m. \end{array} \tag{2.16}$$

This is the primal form of a quadratic optimisation problem, a kind of problem which is routinely solved according to optimisation theory [Bazaraa et al., 1990, 1993]. Its dual form, whose optimal solution is related to the optimal solution of the original problem, is given by

$$\begin{array}{rc} \max & \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i^T \cdot x_j \\ \text{subject to:} & \alpha_i \geq 0, i = 1, 2, \ldots, m \\ & \sum_{i=1}^m \alpha_i y_i = 0 \end{array} \tag{2.17}$$

where $\alpha_i \geq 0$ are Lagrange multipliers.

If $\alpha^{opt} = (\alpha_1^{opt}, \alpha_2^{opt}, \ldots, \alpha_m^{opt})$ is the optimal solution of the dual problem, it can be proved that the vector $\sum_{i=1}^m \alpha_i^{opt} y_i x_i^T$ is normal to the maximal margin hyperplane, and thus, corresponds to w^{opt}.

In addition, the optimal solution given by α^{opt} and w^{opt} must satisfy the Karush-Kuhn-Tucker complementarity conditions [Cristianini and Shawe-Taylor, 2000]

$$\alpha_i^{opt}[y_i \cdot (w^{opt} \cdot x_i + b) - 1] = 0,\ i = 1, 2, \ldots, m. \tag{2.18}$$

These conditions imply that $\alpha_i^{opt} > 0$ only for values of i such that x_i lies relatively close to the maximal margin hyperplane. For this reason, such x_i is called support vector. Since the sign of the coefficient of x_i is given by the classification y_i, the α_i^{opt} are positive values proportional to the influence of x_i on the optimal solution of the problem.

The Karsh-Kuhn-Tucker complementarity conditions also mean that for values of i such that $\alpha_i^{opt} > 0$,

$$b = y_i - w^{opt} \cdot x_i. \tag{2.19}$$

For numerical stability, b is usually calculated as an average

$$b = -\frac{1}{2}\left(\max_{x_i; y_i=-1; \alpha_i^{opt}>0} w^{opt} \cdot x_i + \min_{x_i; y_i=1; \alpha_i^{opt}>0} w^{opt} \cdot x_i \right). \tag{2.20}$$

Soft Margin Support Vector Machines

When the two classes are not linearly separable, the condition for the optimal hyperplane in Equation 2.10 can be relaxed by including an additional error term $\xi_i \geq 0$

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad (i = 1, \cdots, m). \tag{2.21}$$

To achieve a minimum error, ξ_i should be minimised as well as $\|w\|$, and the objective function becomes

$$\begin{array}{rl} \min & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^m \xi_i \\ \text{subject to:} & y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \text{and} \\ & \xi_i \geq 0, \quad (i = 1, \cdots, m) \end{array} \tag{2.22}$$

where C is a regularisation parameter that controls the trade-off between maximising the margin and minimising the training error. Small C values emphasise the margin, while large C values may tend to overfit the training data.

The dual form of this optimisation problem, which is solved for α_i, with $i = 1, 2, \ldots, m$, as a quadratic programming problem with slack variables ξ_i, is

$$\begin{array}{rl} \max & \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i^T x_j \\ \text{subject to:} & 0 \leq \alpha_i \leq C, i = 1, 2, \ldots, m, \\ & \sum_{i=1}^m \alpha_i y_i = 0. \end{array} \tag{2.23}$$

As in the separable data case, if α^{opt} is the optimal solution, the vector $\sum_{i=1}^m \alpha_i^{opt} y_i x_i^T$ realises the maximal margin hyperplane, and thus, corresponds to w^{opt}. The parameter b can be calculated from any support vector, that is, any x_i such that $0 \leq \alpha_i^{opt} \leq C$, as $b = y_i - w^{opt} \cdot x_i$.

Non-linear Support Vector Machines

Linear SVMs converge only for linearly separable data. However, a remarkable property of SVMs is their ability to learn non-linear functions, such as polynomial, radial basic, and sigmoidal functions, which increases their computational power. This is achieved by mapping the data into another space, normally a space of higher dimension, where the classes are linearly separable.

Given an instance x in the original feature space $\Omega_X \subseteq \mathbb{R}^{n_1}$, a space $F \subseteq \mathbb{R}^{n_2}$, and a mapping $\phi : \Omega_X \to F$, the projection of x into the new space F is

$$\phi(x) = (\phi_1(x), \ldots, \phi_{n_2}(x)). \tag{2.24}$$

This is equivalent to mapping the input space Ω_X into a new space $F = \{\phi(x) | x \in \Omega_X\}$. The function

ϕ is called a *kernel-induced* implicit mapping. Kernel-methods preprocess the data by some non-linear mapping ϕ and then apply the linear algorithm described in Equation 2.23, but in the image space of mapping ϕ.

The Kernel Trick

A kernel is a function that takes two vectors x_i and x_j as arguments and returns the value of the inner product of their images $\phi(x_i)$ and $\phi(x_j)$

$$K(x_i, x_j) = \phi(x_i)^T \cdot \phi(x_j).$$

As only the inner product of the two vectors in the new space is returned, the dimensionality of the new space is not relevant.

The learning algorithm in the kernel space can be obtained by replacing all inner products in the learning algorithm in the original space with the kernels

$$h(x) = w^{opt} \cdot \phi(x) + b = \sum_{i=1}^{m} \alpha_i^{opt} y_i K(x_i, x) + b. \tag{2.25}$$

The hyperplane with maximal margin, $w^{opt} \cdot x + b = 0$, can be formulated in terms of the optimal solution α^{opt} of the following optimisation problem

$$\begin{array}{rc} \max & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ \text{subject to:} & 0 \le \alpha_i \le C, i = 1, 2, \ldots, m, \\ & \sum_{i=1}^{m} \alpha_i y_i = 0, \end{array} \tag{2.26}$$

with

$$w^{opt} = \sum_{i=1}^{m} \alpha_i^{opt} y_i \phi(x_i)^T.$$

and

$$b = y_k - w^{opt} \cdot \phi(x_k) \tag{2.27}$$

found as usual from any support vector x_k.

As the vectors x_i appear only in inner products in both the decision function and the learning law, the mapping function $\phi(x)$ does not need to be explicitly specified. Instead, all we need is the inner product of the vectors in the new space.

There are number of standard kernel functions, including linear, polynomial, and gaussian or radial basis function (RBF):

$$K(x_i, x_j) = \phi(x_i)^T \cdot \phi(x_j) = \begin{cases} x_i^T \cdot x_j & \text{linear} \\ (x_i^T \cdot x_j)^d, \ d \in \mathbb{N} & \text{polynomial of degree d} \\ e^{\frac{(-||x_i - x_j||^2}{\sigma^2}}, \ \sigma \in \mathbb{R} & \text{gaussian with parameter } \sigma. \end{cases}$$

Proximal Support Vector Machines

As opposed to soft margin SVMs, which classify instances by assigning them to one of two disjoint half-spaces, proximal Support Vector Machines (PSVMs) [Fung and Mangasarian, 2001] classify instances by assigning them to the closer of two parallel planes. The PSVM algorithm requires the solution of a single system of linear equations and can be easily implemented in an efficient way. PSVM classifiers have been tested on publicly available datasets, with comparable test set correctness to that of standard SVM classifiers and considerably faster computational times.

Consider the problem of classifying the instances $x_1, x_2, .., x_m \in \Omega_X \subseteq \mathbb{R}^n$, according to membership of each point in two classes, $+1$ and -1. The decision function of a standard Support Vector Machine with a linear kernel is determined by the hyperplane that separates positive and negative training examples with maximum margin. From Equation 2.22 it can be deduced that if the training data is represented by a $m \times n$ matrix A and the labels are given by the elements in the diagonal $m \times m$ matrix D, the problem of finding the optimal separating hyperplane can be formulated as the following optimisation problem

$$\begin{aligned} \min_{(w,\lambda,\xi)\in\mathbb{R}^{m+1+n}} \quad & \tfrac{1}{2}\|w\|^2 + C\cdot\xi\cdot e \\ \text{subject to:} \quad & D(A\cdot w^T - \lambda e) \geq e - \xi^T, \\ & \xi \geq 0, \end{aligned} \tag{2.28}$$

where C is a constant that controls the amount of admissible errors $\xi = [\xi_1, \xi_2, \ldots, \xi_m]$, e is a column vector whose m components are $1s$, w is a vector normal to the separating hyperplane, and the constant λ determines its location relative to the origin.

Now, consider solving instead the following problem, which has shown a performance equivalent to that of the classical formulation [Lee and Mangasarian, 2000-2001].

$$\min_{(w,\lambda,\xi)\in\mathbb{R}^{m+1+n}} \quad \tfrac{1}{2}(\|w\|^2 + \lambda^2 + C\cdot\|\xi\|^2) \tag{2.29}$$

$$\text{subject to:} \quad D(A\cdot w^T - \lambda e) \geq e - \xi^T. \tag{2.30}$$

Negative error components ξ_k do not minimise the objective function, and therefore will not be part of the optimal solution, making non-negativity constraints on ξ redundant; in fact, setting $\xi_k = 0$, $k = 1, 2, \ldots, m$, will decrease the objective function without affecting the corresponding inequality constraint.

Replacing the inequality constraint by an equality constraint, results in a significant change in the nature of the optimisation problem [Fung and Mangasarian, 2001]

$$\min_{(w,\lambda,\xi)\in\mathbb{R}^{m+1+n}} \quad \tfrac{1}{2}\left(\|w\|^2 + \lambda^2 + C\cdot\|\xi\|^2\right) \tag{2.31}$$

$$\text{subject to:} \quad D(A\cdot w^T - \lambda e) = e - \xi^T. \tag{2.32}$$

The exact solution for this problem can be written explicitly in terms of the training data. The planes $w \cdot x - \lambda = \pm 1$ can be thought of as *proximal* planes, around which the points of each class are clustered. The distance between the two planes is given by $\frac{2}{\|w\|}$. Therefore, one of the terms in objective function is the reciprocal of the square of the L_2-norm distance between the planes. The proximal planes are pushed as far as possible by the term $\|w\|^2 + \lambda^2$.

In contrast to standard SVMs, PSVM classifiers can be obtained by solving a single system of linear equations, which makes them fast and easy to implement. An additional advantage is the possibility of writing an explicit exact solution in terms of the problem data.

2.3.2 Markov chains

One of the most popular methods applied to the splice site prediction problem are Markov chains. A discrete-time Markov chain [Durbin et al., 1998] is a sequence of random variables $(X_0, X_1, X_2, \ldots)$ taking discrete values in $\{a_0, a_1, \ldots, a_N\}$, such that for each possible sequence of values $(a_{k_i})_{i\in\mathbb{N}_0}$ with $k_i \in \{0, 1, \ldots, N\}$, the following identity holds

$$P(X_i = a_{k_i}|X_0 = a_{k_0}, X_1 = a_{k_1}, \ldots, X_{i-1} = a_{k_{i-1}}) = P(X_i = a_{k_i}|X_{i-1} = a_{k_{i-1}}). \tag{2.33}$$

The notation is usually simplified to

$$P(X_i|X_0, X_1, \ldots, X_{i-1}) = P(X_i|X_{i-1}). \tag{2.34}$$

For $i = 1, 2, \ldots$, the conditional distribution $P(X_i|X_0, X_1, \ldots, X_{i-1})$ characterises the Markov chain, and is known as *transition probability* of the process. The first observation, X_0, is the *initial state* of the process, and $P(X_0)$ is referred to as *initial distribution.* The value a_{k_i} is the *state* of the process at time i, and in general, the range of the random variables is known as *state space.*

The identity in Equation 2.34 is also called *Markov property.* The expression on the right side of the equal sign defines the *order* of the chain, which is the number of preceding states that determine the transition probabilities; therefore, Equation 2.34 corresponds to a first-order Markov chain. In general, for $k \in \mathbb{N}_0$, the conditional distribution of a *kth*-order Markov chain is given by

$$P(X_i|X_0, X_1, \ldots, X_{i-1}) = P(X_i|X_{i-k}, X_{i-k+1}, \ldots, X_{i-1}). \tag{2.35}$$

By applying the definition of conditional probabilities many times, the probability of a sequence $(X_1, X_2, \ldots, X_l)$ can be described as

$$P(X_0, X_1, \ldots, X_l) = P(X_l|X_0, \ldots, X_{l-1})P(X_{l-1}|X_0, \ldots, X_{l-2}) \ldots P(X_1|X_0)P(X_0), \tag{2.36}$$

and the probability of a sequence $(X_0, X_1, X_2, \ldots, X_l)$ given a certain *kth*-order model is described by

$$P(X_0, X_1, \ldots, X_l|model) = P(X_0, \ldots, X_k) \cdot \prod_{i=k}^{l} P(X_i|X_{i-1}, \ldots, X_{i-k}). \tag{2.37}$$

In the particular case of a first-order Markov chain, the probability of a sequence $(X_i)_{i=0}^{l}$ is reduced to

$$P(X_0, X_1, \ldots, X_l) = P(X_0) \cdot \prod_{i=1}^{l} P(X_i|X_{i-1}). \tag{2.38}$$

The most common assumption is that the transition probabilities are independent from time (i.e., the position in the sequence); the model is then called *homogeneous* Markov chain. In contrast, if the transition probabilities vary over time, the model is an *inhomogeneous* Markov chain.

The states of a Markov chain that models a nucleotide sequence are given by each of the four nucleotides A, C, G and T (or U). The transition probabilities determine the probability of a certain nucleotide following another nucleotide, i.e., one state following another state. For example, a

$2nd$-order Markov chain uses dinucleotides, and keeps track of which nucleotides follow each possible dinucleotide. The transition probabilities can be estimated by calculating the frequency of the transitions from each dinucleotide to each nucleotide in the training sequence, i.e., by counting how many times a certain dinucleotide $(AA, AC, AG, \ldots, TT)$ is followed by A, C, G, and T, respectively. The probability $p_{x_i S_j}$ to observe nucleotide x_i at position m given that a certain sequence of nucleotides S_j has been observed at positions $m-k, \ldots, m-2, m-1$ is called transition probability, and the matrix whose $(i,j)'th$ element is $p_{N_i S_j}$ is the transition matrix associated with the system.

Markov chain classifiers

Besides their ability to model nucleotide sequences, Markov chains can also be used as classifiers. In order to do that, we construct a *positive model*, which in our case represents real splice site sequences, and a *negative model*, which corresponds to pseudo splice site sequences.

For a given nucleotide sequence $(x_i)_{i=0}^{l}$, with $x_i \in \{A, C, G, T\}$ for each $i = 0, 1, \ldots, l$, we can call $p^{+}_{x_{i-1}x_i}$ the transition probability from the state x_{i-1} observed at time step $i-1$ to the state x_i observed at time step i, estimated in the *positive model*; in an analogous manner, $p^{-}_{x_{i-1}x_i}$ is the transition probability from the state x_{i-1} observed at time step $i-1$ to the state x_i observed at time step i, estimated in the *negative model*. Applied to the splice prediction problem, sequence x can be classified as either a real splice site or a pseudo splice site, by calculating the log-odds ratio

$$score(x) = \log \frac{P(x|positive\,model)}{P(x|negative\,model)} = \sum_{i=1}^{l} \log \frac{p^{+}_{x_{i-1}x_i}}{p^{-}_{x_{i-1}x_i}}. \tag{2.39}$$

Although this description applies to first-order Markov chain classifiers, it can be easily generalised for higher-order Markov chain classifiers.

Position-specific Weight Matrix Models

Position-specific weight matrix models (WMMs, introduced by Staden [1988]) contain the frequencies of motifs occurring in a sequence. WMMs assume that the probability of observing a certain nucleotide at any position does not depend on the occurrence of a nucleotide at any other position in the sequence, i.e., that the probabilities are independent. The simplest model counts the occurrences of each nucleotide at each position, and then divides them by the total number of sequences to create a probability; higher-order models consider the frequency of occurrence of dinucleotides, trinucleotides, etc.

WMMs are actually *inhomogeneous* $0th$-order Markov chains; the transition probabilities are reduced to $P(X_i|X_0, X_1, \ldots, X_{i-1}) = P(X_i)$, but depend on the time step i.

WMMs can be used to score new sequences to determine how likely they are to be instances of the target motif. This can be calculated as the joint probability of seeing the nucleotides observed at each position in the new sequence. If $P(X_i = y)$ is the probability of observing a certain nucleotide $y \in \{A, C, G, T\}$ at position i of the sequence, the probability of a nucleotide sequence $x = (x_1, x_2, \ldots, x_l)$, where $x_i \in \{A, C, G, T\}$ for each $i = 1, 2, \ldots, l$, is given by

$$P(x|model) = \prod_{i=1}^{l} P(X_i = x_i), \tag{2.40}$$

where x_i is the nucleotide observed in the sequence at position i.

One problem of WMMs concerns zero-probabilities, which occur when a nucleotide is not represented in any of the sequences in the training set. This is normally solved by adding a small value (*pseudocount*) to all positions. In addition, as the four nucleotides do not occur with equal probability, in order to use WMMs as classifiers the probabilities $P(X_i)$ are typically divided by a background probability (negative model).

Position-specific Weight Arrays

Weight array methods (WAMs) were introduced by Zhang and Marr [1993], as a variant of position-specific weight matrices that tries to capture possible dependences between adjacent positions in a sequence. These models are essentially inhomogeneous high-order Markov chains, and zero-order models correspond exactly to WMMs. As mentioned before, in an inhomogeneous Markov chain, the transition probabilities in Equation 2.35 are functions of the time, i.e., vary with the value of i.

2.3.3 Maximal dependence decomposition models

Maximal dependence decomposition models (MDD) are a method developed by Burge and Karlin [1997] with the aim of capturing most significance dependences between (not necessarily adjacent) positions.

The method involves calculating a measure of correlation between all pairs of positions in the sequence; if no significant dependences are detected, the positions are modelled by simple WMMs; if mostly significant dependences between adjacent positions are found, the positions are modelled by WAMs. This process results in several independent models, which are combined into one in order to model the whole sequence.

2.3.4 Probabilistic or Bayesian networks

A probabilistic or Bayesian network is a directed acyclic graph, whose nodes represent random variables and whose arcs represent the joint distribution of the variables they connect.

A Bayesian network can be interpreted as the joint probability distribution of the variables represented in the network. If there is an arc from node X to node Y, then X is said to be a parent of Y. Let $parents(X_i)$ be the parents of the node X_i. Then the joint distribution for $X_0, X_1, \ldots, X_l$ is

$$P(X_0, X_1, \ldots, X_l) = \prod_{i=0}^{l} P(X_i | parents(X_i)), \tag{2.41}$$

where every variable in a probabilistic tree network has at most one parent. Each variable in a probabilistic tree network is allowed to depend on at most one other variable; however, a single variable can influence more than one variable.

Bayesian Networks can infer joint probability distributions based on the observation of single variables; this is particularly useful when a relationship between variables is suspected to be important but our understanding of that relationship is incomplete.

Given a positive and a negative model, the classification of an instance x can be decided according to

$$\frac{P(x|positive\,model)}{P(x|negative\,model)} \tag{2.42}$$

and a threshold value for this ratio.

Bayesian Networks constitute a generalisation of first-order Markov chains, and are theoretically guaranteed to match or outperform such models.

Naïve Bayes classifiers

Naïve Bayes classifiers (NBMs) are a special case of Bayesian networks, in which each variable X_i has a common only parent Y_j, which in turn has no parents. Naïve Bayes models are so named because of their *naïve* assumption that all variables X_i are mutually independent given a variable Y_j.

The joint distribution is then given by

$$P(Y_j, X_0, X_1, \dots, X_l) = \prod_{i=0}^{l} P(X_i|Y_j). \tag{2.43}$$

In a classifier, the variable Y_j represents the class.

2.4 Feature Selection Methods

Feature or subset selection is the process wherein a subset of the features available from the data is selected for the application of a learning algorithm. It can be applied in order to increase the efficiency of the computation, or to reduce over-fitting, which occurs when the data is limited with regards to its dimension.

We applied and studied the influence on the performance of SVMs of three different feature selection methods:

- Weight score filter
- Recursive Feature Elimination-SVM
- L_0-norm algorithm

2.4.1 Weight score filter

The weight score filter (WSF) [Furey et al., 2000] ranks variables according to their correlation to the available classes, for $j = 1, \dots, n$

$$w_j = \frac{\mu_j^+ - \mu_j^-}{(\sigma_j+)^2 + (\sigma_j-)^2}, \tag{2.44}$$

where μ_j^+ and μ_j^- are the mean values of feature j for the class $+1$ and -1, respectively, and σ_j+ and σ_j- are the standard deviations. Large w_j values indicate a strong correlation with either the positive or the negative class, i.e., significance.

The underlying assumption is that features are independent. If such assumption is not met, the method could lead to the selection of redundant features and the omission of features that are not individually important, but complementary to each other.

2.4.2 Recursive Feature Elimination-SVM

The Recursive Feature Elimination-SVM (RFE-SVM) algorithm was proposed by Guyon et al. [2002] and estimates weights according to the optimal solution of a linear discrimination problem, which is solved using a Support Vector Machine. The computational cost of this algorithm is a function of the number of variables, because every time a variable is removed a new SVM must be trained. The selection procedure progressively eliminates features with weak weights, obtained by training a linear SVM.

2.4.3 L_0-norm algorithm

The L_0-norm algorithm [Weston et al., 2003] also employs the weights estimated by a SVM classifier. Nevertheless, the concept is different; in the case of the L_0-norm algorithm, the objective is to maximise the number of variables with weight zero. The authors propose to find the minimum set of variables with non-zero weights, by minimising the L_0-norm of these weights. The problem is solved by an iterative procedure that implies subsequent training of a SVM and multiplication of the training data by the SVM weights.

2.5 Sequence homology

An alignment is an hypothesis of positional homology between the nucleotides (or amino acids, in the case of proteins) of a sequence [Gusfield, 1997]. By comparing two or more sequences, we intend to find evidence about how they have diverged from a common ancestor by a process of mutation and selection. The basic mutational processes are *substitutions*, *insertions* and *deletions*, and the frequencies of such processes depend on natural selection. An alignment between two sequences is obtained by inserting a character into the first sequence, deleting a character from the first string, and substituting or replacing one character in the first sequence with a character in the second sequence; insertions and deletions are referred to as *gaps*. Each operation (substitutions, insertions, deletions) has a score, and the score assigned to the alignment is a function that depends on each of the operations involved and a measure of similarity between the sequences. Since insertions and deletions are comparatively rare, the score for these operations is usually negative, and is often referred to as *gap penalty*.

An alignment can be global or local. Global alignment is the general problem of aligning two or more sequences, first solved by Needleman and Wunsch. Local alignment refers to the restricted alignment of only a smaller part of one or several sequences; the first algorithm to achieve this was proposed by Smith and Watermann.

2.5.1 DIALIGN

DIALIGN is a software program for multiple alignment developed by Morgenstern [1999]. The algorithm constructs pairwise and multiple alignments by comparing whole segments of the sequences, without using any gap penalty, which makes it especially efficient on sequence datasets which are not globally related but share only local similarities.

2.6 Covariation models applied to intron prediction

In molecular biology, structure is normally much more conserved than sequence. In the case of mRNA, this means that when a nucleotide which is implicated in a pair changes, its partner usually changes as well so as to conserve that base pair; this phenomenon is known as compensatory base change.

A common measure for identifying covarying sites is *mutual information*, which is defined as follows. Given two discrete random variables X_i and X_j with ranges Ω_{X_i} and Ω_{X_j} respectively, the mutual information between X_i and X_j is

$$I(X_i, X_j) = \sum_{x \in \Omega_{X_i}} \sum_{y \in \Omega_{X_j}} P(X_i = x, X_j = y) \log_2 \frac{P(X_i = x, X_j = y)}{P(X_i = x) \cdot P(X_j = y)}. \tag{2.45}$$

Mutual information is measured in *bits*.

In a multiple RNA sequence alignment, the mutual information of column i and column j is given by

$$M_{i,j} = \sum_{x \in \{A,C,G,U\}} \sum_{y \in \{A,C,G,U\}} f_{x^i y^j} \log_2 \frac{f_{x^i y^j}}{f_{x^i} f_{y^j}}, \tag{2.46}$$

where f_{x^i} is the frequency of nucleotide x in column i, and $f_{x^i y^j}$ is the joint frequency of dinucleotides x and y in columns i and j respectively.

We evaluated the application of this kind of information to the prediction of splice sites in RNA sequences using the tool COVE [Eddy and Durbin, 1994].

2.6.1 COVE

COVE [Eddy and Durbin, 1994] uses *covariation models* to describe RNA secondary structure patterns. The algorithm learns a model and constructs a consensus secondary structure from initially unaligned example sequences and no prior structural information.

The covariation model is based on an ordered tree, where each node describes columns in a multiple alignment. Specific nucleotides are replaced with *symbol emission probabilities*, corresponding to the 16 possible pairwise nucleotide combinations or four singlet nucleotides. Each node is replaced with a number of *states* that allow for variations in the structure, such as insertions and deletions. *Match states* account for the conserved consensus columns of an alignment, *insert states* account for insertions relative to the consensus, and *delete states* account for deletions with respect to the consensus. The states are connected to each other by *state transition probabilities*. The resulting probabilistic model is a *covariation model* and consists of a number of states M, symbol emission probabilities P, and state transition probabilities T.

The basic operation for the construction and evaluation of a covariation model is the alignment of an RNA sequence to the model and the calculation of the corresponding probability score. A scheme for the construction of the model is presented in Figure 2.4. Given a set of example sequences, COVE finds the covariation model with the maximum likelihood of generating those sequences. For that purpose, both the consensus secondary structure for the model and the optimal values for the state transition probabilities and symbol emission probabilities given that structure have to be determined. The latter problem is solved using expectation maximisation (EM) methods.

In order to choose the consensus secondary structure for the model, COVE uses a dynamic programming algorithm. The mutual information of columns i and j, $M_{i,j}$, represents the expected gain in score from assigning columns i and j to a pairwise state instead of to singlet states. COVE assumes that the consensus secondary structure will be included in the tree that captures as much correlation information as possible, which can be calculated by applying dynamic programming to the calculation of the pairwise mutual information of the columns. The columns of the alignment are assigned to match nodes or insert states of neighbouring nodes depending on the number of symbols in the column, using an arbitrary threshold of 50%.

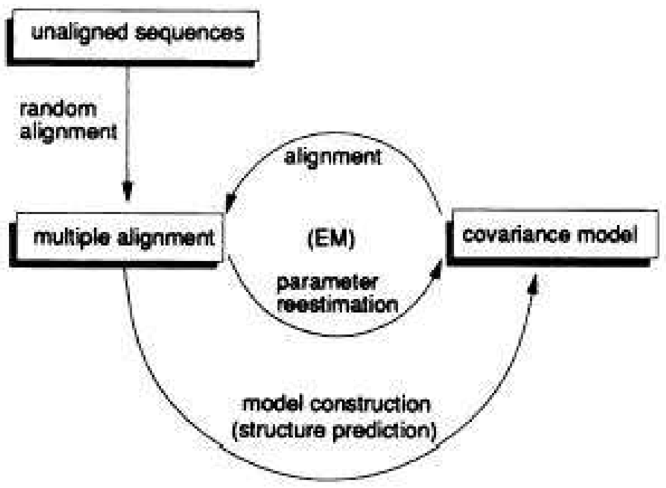


Figure 2.4: The covariation model training algorithm, as described by [Eddy and Durbin, 1994].

2.7 Core methods for gene prediction

2.7.1 Hidden Markov models

Most gene prediction programs use hidden Markov models (HMM, [Durbin et al., 1998]) to model gene structure. A hidden Markov model (HMM) is a statistical model where the system being modelled is assumed to be a Markov chain with unknown (*hidden*) parameters, which are inferred from observable parameters. Hence, a hidden Markov model consists on two stochastic processes, namely, the process of moving between the states and the process of emitting the output symbols. The sequence of state transitions is hidden, but can be observed through the sequence of emitted symbols.

An HMM is defined by the following elements

- S, a set of states $\{s_1, s_2, \ldots, s_N\}$;
- Σ, the output alphabet $v_1, v_2, \ldots, v_M$, which can also be infinite.
- $\Pi = \{\pi(i)\}$, the probability of being at state s_i at time $t = 0$, i.e., $\pi(i) = P(q_0 = s_i)$.
- A, the transition probabilities $\{a_{ij}\}$, where a_{ij} is the probability of entering state s_j at time $t+1$ given that the state at time t is s_i, i.e., $a_{ij} = P(q_{t+1} = s_j | q_t = s_i)$.
- B emission probabilities $\{e_j(k)\}$, where $e_j(k)$ is the probability of observing v_k at time t given that the state at time t is s_j, i.e., $e_j(k) = P(v_k | q_t = s_j)$.

Given the definitions above, the notation of a model is $l = (A, B, \Pi)$. An example for a HMM is shown in Figure 2.5.

In a gene prediction program, the states normally represent gene components, such as promoters, splice-sites, start and stop sites. The states are hidden, but determine the probability of generating particular (observable) nucleotides.

In a *generalised* hidden Markov model (GHMM, [Kulp et al., 1996]) subsequent states have arbitrary (instead of fixed) length distributions; this is useful to model exons and introns, whose lengths appear to be restricted but do not exhibit geometric distributions.

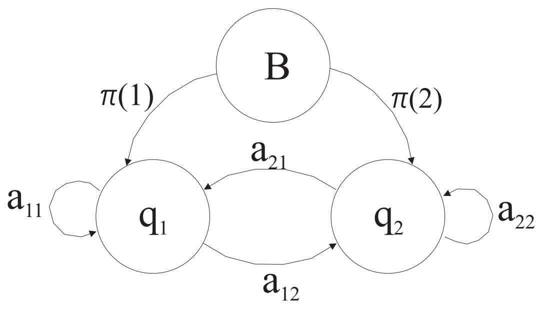


Figure 2.5: Scheme representing a HMM with two transition states q_1 and q_2 and one inital state B. The transition probabilities between the states are also indicated. In turn, each state can produce a symbol k from the alphabet Σ, according to probabilities $\{e_1(k)\}$ and $\{e_2(k)\}$, respectively.

2.7.2 Artificial neural networks

Artificial neural network (ANN) models consist of one or more layers of highly interconnected elements that are linked with weighted connections; they are named after their analogy with the brain's neurons and synapses. The work presented in this thesis does not involve neural networks, but some gene prediction programs rely on them [Snyder and Stormo, 1993].

2.8 Classification criteria

For evaluating the performance of the different classification methods, we employ two criteria

- Area under the Receiver Operator Characteristic curve (AUC), and
- Classification Rate.

Both methods are good indicators of the performance of classification system. However, when the data are strongly unbalanced, as it is in the case of splice sites, the classification rate may be misleading since the all-positive or all-negative classifier may achieve a very good classification rate. In that regard, the AUC represents a more robust criterion.

2.8.1 Receiver Operator Characteristic curves

Receiver Operator Characteristic (ROC) curves were developed in the 1950's as a by-product of research into making sense of radio signals contaminated by noise. In the present, ROC curves are extensively used for the evaluation of machine learning results, and are commonly applied in epidemiology and medical research [Swets, 1988].

In the case of a binary classifier, a ROC curve is a graphical plot of the fraction of true positives (sensitivity) as a function of the fraction of false positives (1-specificity), as its discrimination threshold is varied.

$$Sensitivity = \frac{TP}{TP + FN} \tag{2.47}$$

and,

$$1 - Specificity = 1 - \frac{TN}{TN + FP}, \tag{2.48}$$

where

- TN is the number of true negative predictions,

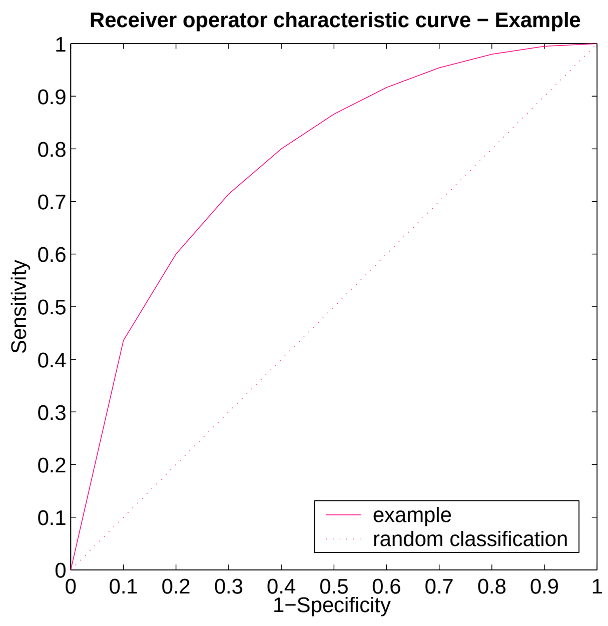


Figure 2.6: Illustration of the use of ROC curves for discrimination.

- TP is the number of true positive predictions,
- FN is the number of false negative predictions,
- FP is the umber of false positive predictions.

In Figure 2.6, the diagonal dotted curve corresponds to a completely random predictor; the score distributions of the positive and the negative examples overlap, and therefore, raising the threshold has no influence on the performance of the method. On the contrary, the best possible prediction method would yield 100% sensitivity for 100% specificity, i.e., the graph would include a point in the upper left corner of the ROC space. The slope of the tangent line at a cut-point gives the likelihood ratio for that value of the test. Finally, the area under the curve (AUC) is a measure of the accuracy of the test.

2.8.2 Classification Rate

The classification rate is simply the fraction of examples that was correctly classified

$$CR = \frac{TP + TN}{TP + FP + TN + FN}. \tag{2.49}$$

2.9 Evaluation of gene prediction methods

The accuracy of gene prediction methods is commonly measured at four different levels [Burset and Guigó, 1996]:

- nucleotide level, which refers to the coding/non-coding nucleotides detected;
- exon level, with respect to the exact prediction of exon boundaries;
- gene level, regarding the predicted gene structure; and

- protein level, with respect to the protein product encoded by the predicted gene.

In addition, the accuracy is expressed through the *sensitivity* and the *specificity* of the tests; however, in this context the definition of specificity refers actually to what is known as *positive predictive value*:

$$Sensitivity = \frac{TP}{TP + FN}, \tag{2.50}$$

and

$$Specificity\ (Positive\ predictive\ value) = \frac{TP}{TP + FP}. \tag{2.51}$$

The *sensitivity* is then the fraction of correct nucleotides, exons or genes over the actual nucleotides, exons or genes, and the *specificity* is the fraction of correct nucleotides, exons or genes over the predicted nucleotides, exons or genes.

Chapter 3

Related Work

3.1 Eukaryotic gene prediction

Although gene prediction is a very active research field, existing gene prediction programs still have important limitations [Mathé et al., 2002, Rogic et al., 2001, Burge and Karlin, 1997, Burset and Guigó, 1996]. More than 90% of the nucleotides can be identified correctly as either coding or non-coding, but the exact boundaries of the exons are still difficult to predict [Stormo, 2000]. In addition, gene prediction is very sensitive to species-specific parameters, which means that every genome needs a dedicated gene finder [Korf, 2004].

The objective of the development of gene prediction programs is to automate or facilitate the identification of genes on new data. For many purposes, having only partially correct predictions can already be helpful to focus and therefore accelerate experiments that can determine true gene structures.

Gene prediction programs are based on two general strategies. *Extrinsic* or *homology-based* methods rely on alignment algorithms to identify homologies with annotated sequences in protein, cDNA or EST databases; this strategy has been extensively used, and with considerable success, but is only able to identify homologous genes, and fails when the target sequence does not present any detectable similarity to previously identified gene sequences. On the other hand, *intrinsic* or *ab initio* methods use pattern recognition techniques to identify signals in the sequences. The combination of both methods has proved to produce stronger results [Mathé et al., 2002, Birney and Durbin, 2000]. A third gene prediction strategy takes advantage of the fact that functional regions are conserved during evolution, identifying potential coding regions based on the homologies found between two or more sequences.

3.1.1 *Ab initio* gene prediction programs

Most *ab initio* gene prediction programs use Hidden Markov Models (HMMs). The gene structure model is divided into submodels or *states* representing gene components, such as promoters, splice-sites, start and stop sites; the transition probabilities between these states are modelled as unobserved (*hidden*) Markov processes, i.e., processes in which the current state depends only on the previous state. The states are hidden, but determine the probability of generating particular (observable) nucleotides.

A particularly popular program that uses generalised Hidden Markov Models is GENSCAN [Burge and Karlin, 1997]. In this program, the structure of the sequence is modelled by generalised

Hidden Markov Models (GHMMs). Signals are modelled by weight matrices (WMMs), weight arrays (WAMs), and maximal dependence decomposition (MDD). Figure 3.1 illustrates the model for genomic regions utilised by GENSCAN.

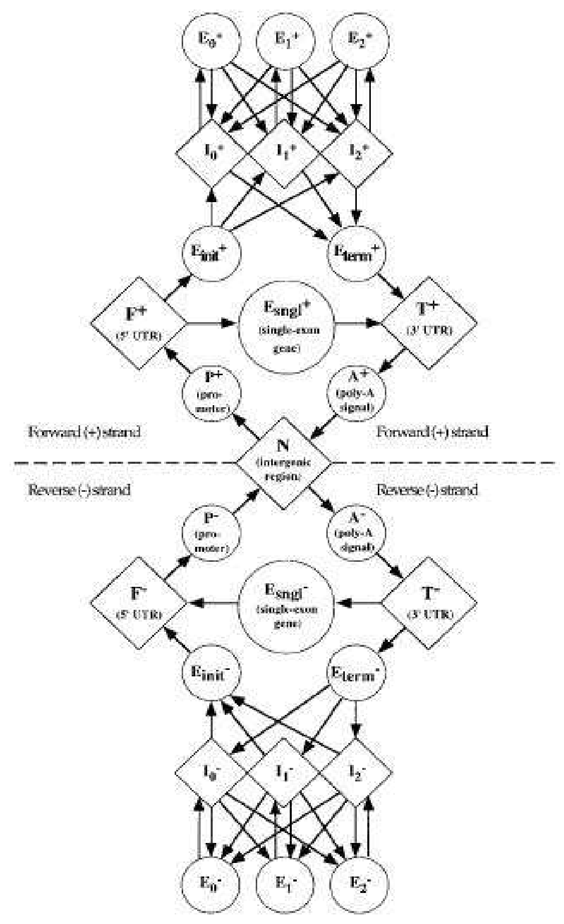


Figure 3.1: GHMM for predicting gene structures, as presented by Burge and Karlin [1997]. Each shape represents a functional unit of a gene or genomic region. Label N corresponds to intergenic regions, P to promoters, F to 5' untranslated regions, E_{sngl} to single exons (intronless genes), E_{init} to initial exons, E_k to internal exons in phase k (the definition of *phase* is related to that of the reading frame), E_{term} to terminal exons, T to 3' untranslated regions, A to polyadenylation signals, and i_k to introns in phase k.

GENSCAN generates a *parse* or sequence of states $\phi = q_1, q_2, \ldots, q_K$ chosen from the set of states $O = \{o_1, o_2, \ldots, o_N\}$, $N \in \mathbb{N}$, with associated lengths $d_1, d_2, \ldots, d_K$ with $K \in \mathbb{N}$. This process generates a nucleotide sequence s.

The parse corresponding to a given nucleotide sequence s of length l starts with the choice of the initial state, q_1, according to an initial probability distribution $\Pi = \{\pi(i)\}$ with $\pi(i) = P(q_1 = o_i)$. Then, a length d_1, is generated depending on the value of q_1, which we will refer to as o_{r_1}, and the corresponding length distribution function $f_{o_{r_1}}$. Conditional on the values of q_1 and d_1, a sequence segment s_1 of length d_1 is generated. The subsequent state q_2 depends on o_{r_1} and on the transition probabilities between the states given in the transition matrix $T = (t)_{ij}$, where $t_{ij} = P(q_{r+1} = o_j | q_r = o_i)$ is the probability of entering state o_j at time $r + 1$ given that the state at time r is o_i. This process is repeated n times, until $\sum_{i=1}^{n} d_i \geq l$; at that time, the final state is appropriately truncated to generate the last section of the sequence. Finally, sequence s is generated by $s_1, s_2, \ldots, s_n$.

Given a sequence s of length l and a parse ϕ_i, the joint probability $P(\phi_i, s)$ is given by

$$P(\phi_i, s) = \pi(q_1)d_1 P(s_1|q_1, d_1) \cdot \prod_{j=2}^{n} t_{q_{j-1}q_j} d_j P(s_j|q_j, d_j). \tag{3.1}$$

Let Φ_l be the set of all possible parses of length l, and S_l be the set of all nucleotide sequences of length l. The conditional probability of a parse $\phi_i \in \Phi_l$ given a sequence $s \in S_l$ can be calculated as

$$P(\phi_i|s) = \frac{P(\phi_i, s)}{P(s)} = \frac{P(\phi_i, s)}{\sum_{\phi_j \in \Phi_l} P(\phi_j, s)}. \tag{3.2}$$

Based on Equation 3.2, and given a probabilistic model for the different genomic regions, GENSCAN determines the parse with the highest likelihood given a sequence s.

The output of GENSCAN gives information about exon location and their probabilistic score, as well as the scores for other sequence features.

Another *ab initio* gene prediction program, based on the generalised Hidden Markov Models first introduced by GENSCAN, is AUGUSTUS [Stanke and Waack, 2003]. This program provides a more accurate method for modelling the length distribution of introns, by combining explicit length modelling (for introns shorter than a few hundred bases) with a geometric distribution (for longer introns).

Gene prediction programs based on HMMs contain one or more states modelling introns. The lengths of such states either respond to an explicit distribution or are modelled by emitting one nucleotide at a time and allowing transitions back to the same state.

Explicit length distributions are accurate, but computationally expensive; considering that the running time of the typical algorithms is at least proportional to the maximal possible length of the states, and that introns can be very long, the explicit modelling of length distributions is in practice infeasible. On the other hand, using states that emits only one nucleotide and allowing transitions back to the state is computationally efficient, but restricts length distributions to *shifted* geometric distribution. AUGUSTUS proposes a compromise solution, by modelling length distributions explicitly up to length d, and using for the rest a geometric distribution. AUGUSTUS intron model is shown in Figure 3.2.

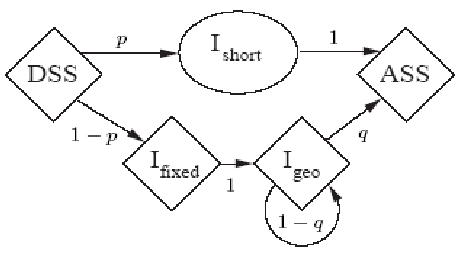


Figure 3.2: Detail of the intron model suggested by Stanke and Waack [2003]. ASS and DSS represent the acceptor splice site, and the donor splice site, respectively. The state I_{fixed} emits a sequence stretch of fixed length d, while the state I_{geo} emits a sequence stretch of variable length, according to a geometric distribution with parameter q. If the intron has length l, and $l \leq d$, the path goes through state I_{short}; otherwise, the path goes first through state I_{fixed}, and subsequently, $l - d$ times through state I_{geo}.

Other ab initio gene prediction programs for eukaryotes include GeneID [Guigó et al., 1992], FGENESH [Salamov and Solovyev, 2000, Solovyev et al., 1995], GeneParser [Snyder and Stormo, 1995, 1993], Genie [Kulp et al., 1996], GenLang [Dong and Searls, 1994], GRAIL [Uberbacher et al., 1996], HMMgene [Krogh, 1997], GeneMark.hmm [Borodovsky and Lukashin, Lomsadze et al., 2005], Morgan [Salzberg et al., 1998], MZEF [Zhang, 1997].

3.1.2 *Homology-based* gene prediction programs

As opposed to the programs presented in the previous section, GenomeScan [Yeh et al., 2001], PROCRUSTES [Gelfand et al., 1996], GeneWise [Birney et al., 2004, Birney and Durbin, 1997] are examples of programs based on homology [Wang et al., 2004].

Because of the course of evolution, it is reasonable to assume that there are strong homologies between genes of related species. Homology-based gene prediction programs rely on this supposal, and examine nucleotide and protein databases looking for similarities to previously identified coding regions. As expected, such approaches can only identify genes that are very similar to already known genes, but the fraction of known genes is growing rapidly.

Homology-based gene prediction programs use homology information in different ways. For example, the strategy of PROCRUSTES consists on exploring all possible exon assemblies and finding the multi-exon structure with the best fit to a related protein. The underlying algorithm is called the *spliced alignment algorithm*; PROCRUSTES considers all possible chains of candidate exons (sequence segments limited by all candidate acceptor and candidate donor splice sites) and finds a chain with maximum global similarity to the target protein; as the number of candidate exons can be very high, they are first filtered to eliminate some false positives.

Another homology-based approach to identify unknown genes is to compare two or more genomes looking for conserved regions, based on the idea of evolutionary pressure preserving functional regions. Programs that make use of synteny between organisms include AGenDA [Rinner and Morgenstern, 2002]. AGenDA relies on cross-species sequence comparison to find potentially functional regions. The algorithm clusters the alignments that DIALIGN [Morgenstern, 1999] has calculated between two sequences, and then searches the boundaries of the resulting clusters for conserved splicing signals using position-specific weight matrices.

A summary of the underlying algorithms used by each program mentioned in this work is presented in Table 3.1. The list is not intended to be comprehensive, but representative.

Regardless of the many gene prediction programs currently available, their performance is limited. Although the results produced by these methods may direct the course of laboratory work, they are far from being definite. Indeed, it seems unrealistic to imagine that complex biological processes such as transcription and translation can be explained merely by the nucleotide sequence [Mathé et al., 2002].

3.2 Splice site prediction

Given that most eukaryotic genes are composed of several exons, the most complex and important subtask in the determination of the exact structure of genes in genomic sequences of higher organisms is the prediction of splice signals [Mathé et al., 2002]. An extensive overview of the splice site prediction problem can be found in Sonnenburg [2002], while a more general review and a comparison of gene and splice site prediction methods is discussed by Mathé et al. [2002] and Zhang [2002].

The prediction of splice sites is primarily based on probabilistic models that look for consensus motifs or patterns in the surroundings of the conserved dinucleotides AG and GT (GU in the pre-mRNA sequence) that characterise splice sites. The models estimate some kind of nucleotide distribution in the neighbourhood of the consensus AG and GT from a sequence data set. The most

Program	Algorithm
GENSCAN	GHMM
AUGUSTUS	GHMM
GeneID	WMMs, HMM
FGENESH	HMM
GeneParser	neural networks, EST homology
Genie	GHMM, protein homology
GenLang	Grammar rule
GRAIL	neural networks, dynamic programming
HMMgene	class HMM
GeneMark.hmm	HMM, dynamic programming
Morgan	decision trees, dynamic programming, Markov chains
MZEF	Quadratic discriminant analysis
GenomeScan	probabilistic models, sequence similarity (e.g. BLASTX hits)
PROCRUSTES	explores all possible exon assemblies and finds the structure with the best fit to a related protein
Genewise	protein similarity
AGenDA	syntenty comparison

Table 3.1: Methods used by some of the available gene prediction systems.

common splice site prediction methods include

- Markov chains;
- position-specific weight matrices, and weight array methods;
- maximal dependence decomposition;
- Bayesian networks, and naïve Bayes classifiers; and
- Support Vector Machines.

One significant difference among all these methods is that while position-specific weight matrices, weight array methods, and maximal dependence decomposition models are usually trained only with positive examples, the remaining methods require both positive and negative examples. All methods require the transformation of the nucleotide sequence into a mathematical vector whose components contain information about particular characteristics of the sequence; the components of these vectors are referred to as *features*; if they are contiguous, the positions in the sequence that are considered for the analysis constitute a *window*, which is denoted here as a close interval, and where position 0 corresponds to the occurrence of the first character of the conserved dinucleotides AG or GT.

Position-specific weight matrices are among the earliest methods used for splice site prediction. Since then, research has been oriented towards the development of methods able to capture statistical dependences between positions in the sequence; examples of these include position-specific weight arrays, maximal dependence decomposition models, Bayesian networks, as well as more complex methods like Support Vector Machines. However, in general, the increase of the complexity does not produce a dramatic improvement in splice site discrimination with respect to simpler models which assume independence between positions [Burge and Karlin, 1997]. The following sections present the most relevant related work done.

3.2.1 Position-specific Weight Matrices and Weight Arrays

Position-specific weight matrix models ([Staden, 1988], WMMs) and weight array methods ([Zhang and Marr, 1993], WAMs) are the standard methods for splice site prediction.

Position-specific weight matrices were developed as a replacement for consensus sequences, because of their ability to indicate the relative importance of each base at each position. The method, introduced by Staden [1984], consists of 2-dimensional arrays of values that represent the probability of finding each of the possible sequence characters at each position with respect to the target signal; the probabilities are estimated using the nucleotide frequencies in 130 acceptor site sequences and 139 donor site sequences, and the windows employed are $[-15, 3]$ for acceptor sites, and $[-4, 8]$ for donor sites. Figure 3.3 depicts the (logarithm of the) score distributions resulting from the application of these matrices to the sequences in the training database.

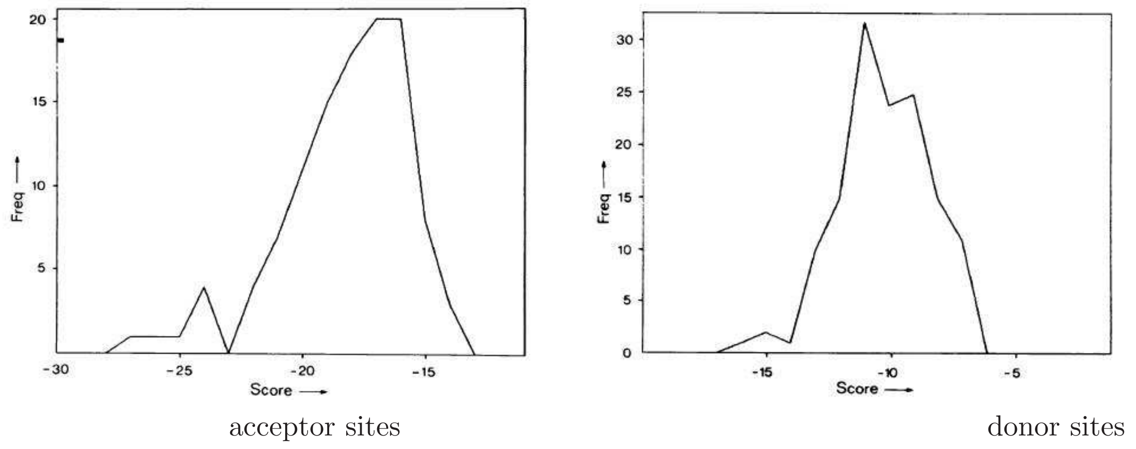


Figure 3.3: Histograms of scores calculated using the matrices proposed by Staden [1984]

WMMs assume that the probabilities of the nucleotides at each position are independent of each other. By contrast, weight array methods were developed to explore and capture dependences between adjacent positions. Zhang and Marr [1993] used conditional probabilities between pairs of contiguous nucleotides at positions $[-4, 8]$ to identify donor sites, and even speculated about the underlying physical mechanism of splicing by relating the discrimination to the biological energetics of the hypothetically intermediate complexes involved in the process; their results indicate the existence of weak correlations, which slightly increase the discriminant power.

The same WAMs were further studied by Salzberg [1997], but using significantly larger training and testing sets and slightly different windows (in the case of human sequences, $[-3, 17]$ for acceptor sites, and $[-14, 2]$ for donor sites). Salzberg [1997] concludes that most of the information required for the discrimination of splice sites is contained in consensus sequence, and that using conditional probabilities consistently improves the results.

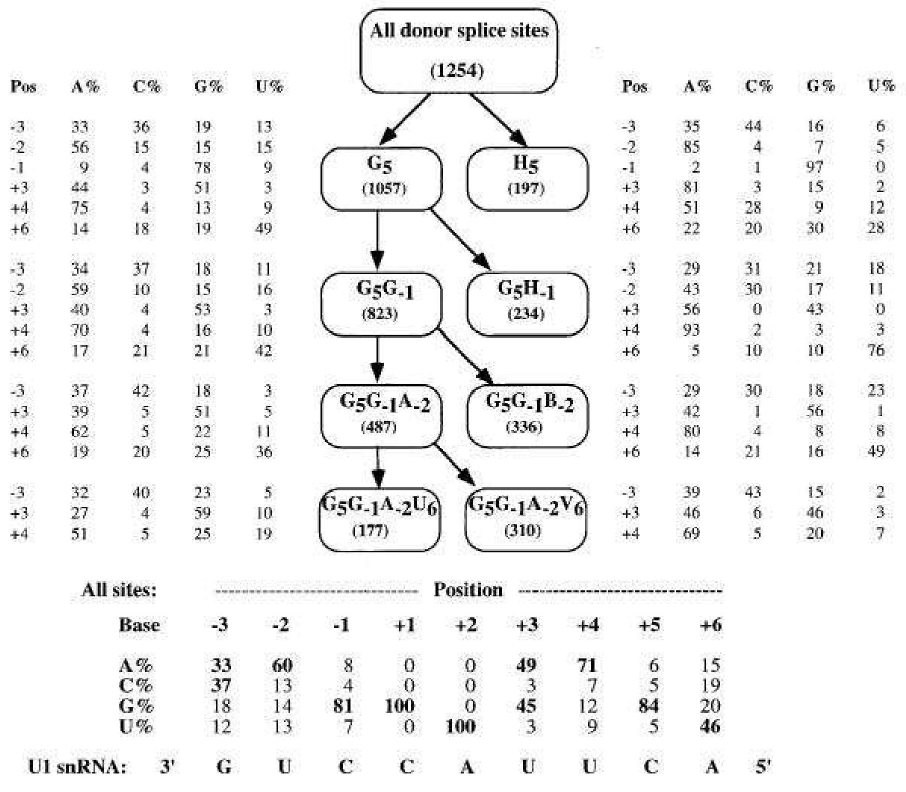
3.2.2 Markov chains

As stated in Chapter 2, both WMMs and WAMs are actually inhomogeneous Markov chains. More general Markov chains are not specifically applied to splice site recognition, but rather are widely used to model base composition to distinguish between coding and non-coding regions [Durbin et al., 1998].

3.2.3 Maximal Dependence Decomposition models

Maximal dependence decomposition models [Burge and Karlin, 1997] represent a further effort to capture the most significant dependences between positions. In this case, the dependences are not restricted to adjacent positions, but may also occur between non-adjacent nucleotides.

Given an alignment of sequences and the corresponding consensus, the method calculates the χ^2 statistic for the consensus indicator variable C_i versus the nucleotide indicator variable X_j for each pair of positions (i, j). The χ^2 test of independence examines whether two categorical variables are correlated. Independent positions are modelled using WMMs, and dependences (correlations) between adjacent positions are modelled with WAMs. For strong dependences between non-adjacent as well as adjacent positions, the method calculates the value $S_i = \sum_{j \neq i} \chi^2(C_i, X_j)$ for each position i, and identifies the value of i, i_{max}, such that this sum is maximal. Subsequently, the data set is divided into a subset of the sequences containing the consensus nucleotide(s) at position i_{max}, and a subset containing the remaining sequences. If possible, each subset of sequences is modelled using WMMs and WAMs; otherwise, the data partition procedure is repeated until some condition is met. This is exemplified in Figure 3.4 for a set of 1254 donor site sequences.



Pos	A%	C%	G%	U%
-3	33	36	19	13
-2	56	15	15	15
-1	9	4	78	9
+3	44	3	51	3
+4	75	4	13	9
+6	14	18	19	49
-3	34	37	18	11
-2	59	10	15	16
+3	40	4	53	3
+4	70	4	16	10
+6	17	21	21	42
-3	37	42	18	3
+3	39	5	51	5
+4	62	5	22	11
+6	19	20	25	36
-3	32	40	23	5
+3	27	4	59	10
+4	51	5	25	19

Pos	A%	C%	G%	U%
-3	35	44	16	6
-2	85	4	7	5
-1	2	1	97	0
+3	81	3	15	2
+4	51	28	9	12
+6	22	20	30	28
-3	29	31	21	18
-2	43	30	17	11
+3	56	0	43	0
+4	93	2	3	3
+6	5	10	10	76
-3	29	30	18	23
+3	42	1	56	1
+4	80	4	8	8
+6	14	21	16	49
-3	39	43	15	2
+3	46	6	46	3
+4	69	5	20	7

All sites:

		Position								
	Base	**-3**	**-2**	**-1**	**+1**	**+2**	**+3**	**+4**	**+5**	**+6**
	A%	**33**	**60**	8	0	0	**49**	**71**	6	15
	C%	**37**	13	4	0	0	3	7	5	19
	G%	18	14	**81**	**100**	0	**45**	12	**84**	20
	U%	12	13	7	0	**100**	3	9	5	**46**
U1 snRNA:	**3'**	**G**	**U**	**C**	**C**	**A**	**U**	**U**	**C**	**A** **5'**

Figure 3.4: MDD model for donor sites [Burge and Karlin, 1997]. Each box represents a subclass of donor splice sites corresponding to a particular pattern of matches and mismatches to the consensus nucleotide(s) at a set of positions in the donor site. H indicates A, C or U; B indicates C, G or U; and V indicates A, C or G. The number of sequences in each subset is given in parentheses. The frequencies (percentages) of each of the four nucleotides at each variable position are indicated for each subclass immediately adjacent to the corresponding box.

When the method is applied to donor sites, the results, presented in Figure 3.4, support the hypothesis that complementarity with the sequence of U1 snRNA or other snRNAs involved in the splicing process might be of primary importance in donor site recognition, i.e., the relevance of position independent information in the nucleotide sequence. However, the results also suggest some subtle dependences [Burge and Karlin, 1997]:

1. $5'/3'$ compensation effect: matches to consensus nucleotides at nearby positions on the same side of the intron/exon junction are positively associated; poor matching on one side of the junction is often compensated by stronger matching on the other side;
2. adjacent base-pair effect;
3. G is preferred at position $+3$ only for a subset of sequences.

3.2.4 Bayesian networks and naïve Bayes classifiers

Also Bayesian networks, including naïve Bayes classifiers(NBMs, [Domingos and Pazzani, 1997]), have been explored as accurate probabilistic models that are able to capture long-distance dependences between non-adjacent bases through a tree structure of conditional probabilities [Cai et al., 2000].

In this case, the model is chosen from a family of Bayes tree networks. Given to random variables X_i and X_j representing the nucleotide composition of the sequence at positions i and j and whose possible values are Ω_{X_i} and Ω_{X_j} respectively, the algorithm starts by computing the mutual information $M(i,j) = \sum_{x\in\Omega_{X_i}} \sum_{y\in\Omega_{X_j}} P(X_i = x, X_j = y) \log_2 \frac{P(X_i=x,X_j=y)}{P(X_i=x)P(X_j=y)}$ between each pair of positions i and j. Subsequently, a weighted graph is constructed, with one node for each position i in the sequence, represented by a random variable X_i. For every two positions i and j in the sequence, the edges between X_i and X_j have associated weights $M(i,j)$. Finally, the algorithm constructs a maximum spanning tree (i.e., an acyclic subgraph containing all nodes of the graph, for which the sum of the edge weights is maximised), and orients the tree by choosing an arbitrary root; for each pair of variables X_i and X_j, such that X_i is the parent of X_j, the probability $P(X_j|X_i)$ can be approximated by the frequencies observed in the data. Such a construction maximises the probability of observing the training data given the model.

The results presented by Cai et al. [2000] corroborate the study by Burge and Karlin [1997], showing that simple first-order Markov chains perform surprisingly well on the splice site prediction problem.

Based on the conclusions of the previous sections, we present a method to improve the performance of high-order position-specific weight matrices on the splice site prediction problem. Higher-order models should in principle allow us to capture complex interactions between the nucleotides in the sequence, but have a number of limitations associated with high state-space complexity and reduced coverage. The method proposed here is able overcome the problems originated by small sample sizes, based on the assumption that probabilities of adjacent positions in the sequence are not completely independent. By means of a Gaussian smoothing operator, we estimate the probability of observing a k-mer at certain position conditioned on the contribution of several neighbouring positions; briefly, this is implemented as follows.

Given a k-mer $q \in \Sigma^k = \{A,C,G,T\}^k$, $k \in \mathbb{N}$, and a sequence of nucleotides $X_1, X_2, X_3, \ldots, X_l$ of length l, with $X_i \in \Sigma = \{A,C,G,T\}$, $i = 1,2,\ldots,l$, the probability of observing k-mer q at position i in the sequence is estimated from the training data as

$$p_i^q = \frac{1}{c}\sum_{j=1}^{l} f_j^q e^{-\frac{(i-j)^2}{2\sigma^2}}, \tag{3.3}$$

where f_j^q is the frequency of occurrence of k-mer q at position j for $j = 1,2,\ldots,l$, σ is the parameter controlling the degree of smoothing, and c is an appropriate normalisation constant. These probabilities define our smoothed position-specific weight matrix model of order k. In addition to a model generated with sequences of known splice sites, we create a *negative model* with sequences that contain the consensus dinucleotides AG or GT but do not correspond to real splice sites.

3.2.5 Support Vector Machines

Weight matrix models (WMMs) and weight array matrices (WAMs), the most popular methods employed for splice site prediction, require the selection of the information sources by hand; by contrast, machine learning techniques have the advantage of inferring an optimal classifier from the training data. Support Vector Machines (SVMs) represent an example of such techniques.

Regarding the literature about prediction of splice sites using SVMs, there have been approaches that focus on how to improve their performance when they are used as classifiers, and others that rather use them as means for the extraction of biological information. Considering that simple systems like WMMs already perform very well on splice site prediction, we propose an approach with a good predictive power but actually aimed at capturing features of relevant biological meaning.

Baets et al. [2002] and Saeys et al. [2004] studied how feature selection methods can improve the predictive performance of SVMs on a data set for the model plant *Arabidopsis thaliana*. Both articles consider donor and acceptor splice sites as two independent problems, and their main objective is the design and evaluation of efficient feature selection algorithms for high-dimensional data.

Baets et al. [2002] explore a wrapper-based feature subset selection method with the purpose of reducing the computational cost by selecting relevant, not necessarily adjacent, nucleotide positions for splice site prediction. Relevant features are identified based on the performance of naïve Bayes classifiers and Support Vector Machines (in combination with Linear, Polynomial and Gaussian kernels), with a modification of the feature subset selection algorithm proposed by Guyon et al. [2002]. Each sequence is represented by features that indicate the presence or absence of one of the four nucleotides at a certain position in the window $[-50, +50]$, where 0 corresponds to the occurrence of the first character of the consensus dinucleotides AG and GT; this results in 400 binary features. Training and testing were accomplished in a 10-fold cross-validation scheme, with equally sized partitions containing as many true splice sites as false (or *pseudo*) splice sites. Under this setup and in contrast to naïve Bayes classifiers, SVMs do not perform well with few features. The results of this work confirm that the most relevant information is the presence or absence of certain nucleotides 3 to 4 positions upstream and downstream the splice sites, and suggest that position invariant patterns may also play an important role. There are significant differences between the feature subsets selected by the four different methods, which could be attributed to the existence of many correlated features.

Saeys et al. [2004] present a second application of SVMs on splice site prediction, in this case, combined with a feature ranking system based on estimation of distribution algorithms (EDAs). EDAs are a class of genetic algorithms that generates new population members based on properties inferred by sampling and estimating properties of the population distribution. The EDA starts by generating an initial population of instances, and then runs iteratively, selecting a subset of individuals according to an evaluation method (in this work, SVMs and Naïve Bayes classifiers), until the termination criteria are met. The feature subset selection by EDA was compared to sequential backward elimination (SBE) and to a method based on Markov blankets, which should be able to capture feature interactions. SBE starts with the complete feature set and iteratively discards features, by training and testing models which leave out each feature once in each model; for m features, this implies $\frac{m(m+1)}{2}$ models; the second method involves the calculation of a correlation matrix. The experiments were done using a 10-fold cross-validation scheme; the partitions were equally sized, but contain approximately

6 times more negative examples (false or pseudo splice sites) than positive examples (true splice sites). Sequences in the window $[50, 50]$ were represented by a) position-dependent information about the presence or absence of nucleotides (400 binary features), b) position-dependent information and position-independent information about the presence or absence of words of length 3, or *trimers* (400 + 128 binary features), or c) additional position-dependent information about the presence or absence of words of length 2, or *dimers* (400 + 128 + 1568 = 2096 features). The results that Saeys et al. [2004] obtained using naïve Bayes classifiers indicate that the data sets contain many irrelevant or correlated features, as many of them can be eliminated without a big impact on the classification performance. The most complex sequence representations produced the best results. EDA performed better than the two reference methods. According to this paper, the most important features for distinguishing true acceptor sites from pseudo sites are nucleotides and dinucleotides in the neighbourhood of the conserved AG; another strong feature is a pyrimidine-rich region upstream of the acceptor site; in addition, TG-dinucleotides upstream of the acceptor were found to be particularly abundant in the sequences of real acceptor sites. The conclusions for donor splice sites were similar; the most relevant features identified were nucleotide occurrences within the consensus region; position-independent trimers effectively captured the T-richness of introns characteristic of *Arabidopsis*; finally, guanosine was found to be under-represented downstream of real donor site sequences, except at position +3, where it was over-represented.

Although the previous two articles concern feature selection algorithms and are oriented towards gaining an insight on the underlying biological process, they are centred on the optimisation of the performance of the SVMs. The sequence representations are not as comprehensive as they could be, and the ratio between splice sites and pseudo splice sites is not adjusted to real conditions. However, they show that feature selection algorithms, in particular those which make use of the output of SVMs as selection criterion, can provide a framework for studying biological processes.

Finally, Zhang et al. [2003] proposed to discover information that might be used by the cell to recognise splice sites by identifying the sequence features that allow SVMs to distinguish between real and pseudo exons, and verified their findings through statistical tests. The evaluation was done on about 2200 real and 2300 pseudo exons from a human sequence database. The exons and their flanking regions (*flanks*) were divided into five non-overlapping components, whose contribution to the performance was evaluated both independently and in combination with other components.

- upstream flank (positions −64 to −15 with respect to the acceptor site)
- acceptor splice site
- exon body (+2 with respect to the acceptor site to −4 regarding the acceptor site)
- donor splice site
- downstream flank (positions +7 to +56 with respect to the donor site)

The sequences of the flanks and exon bodies were represented by the occurrence or absence of words of lengths between 4 and 7 (4^k features, where k is the length of the words). Acceptor and donor sites were represented by position-dependent combinations of three and two nucleotides, respectively. The optimal results corresponded to the sequence representation including all five components, while the omission of some of them significantly compromised the performance (see Figure 3.5). Exons bodies confirmed to be relevant for the distinction between real and pseudo exons, but specific signals for splicing were difficult to identify. The identification of discriminative features was accomplished by Recursive Feature Elimination [Guyon et al., 2002], based on the weights produced by the SVM. Regarding the flanks, the best results were obtained with words of length 4 or 5 (*4-mers* and *5-mers*) and a degree-2 polynomial kernel. The performance varied significantly when one of either flanks was left out or when longer words were used, suggesting that this representation captures information which is specific for splice site sequences. By partitioning the original 50-nucleotide-long windows into shorter windows, and observing the sum of the weights of the most relevant 5-mers for each small window, Zhang et al. [2003] concluded that most important signals are within 50 nucleotides of the exon borders. The 5-mers found were clustered according to their homology:

- pyrimidine-rich sequences, which occur with high rates in the upstream flank of real exons and are part of the polypyrimidine track that is known to precede the acceptor site, and seem to be less prevalent in the downstream flank of pseudo exons.
- 5-mers compatible with the branch point in the upstream flank.
- G-rich sequences in the downstream flank, in part seen as a compensatory effect to the polypyrimidine track upstream of the acceptor site.
- C-rich sequences downstream and upstream of the exons, distinct from the polypyrimidine tract.
- TG-rich sequences peak immediately upstream of the exon.
- other.

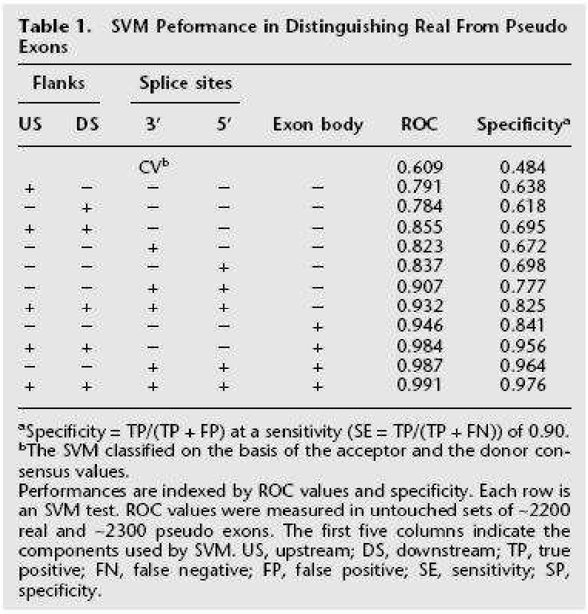


Table 1. SVM Peformance in Distinguishing Real From Pseudo Exons

Flanks		Splice sites				
US	DS	3′	5′	Exon body	ROC	Specificity[a]
		CV[b]			0.609	0.484
+	−	−	−	−	0.791	0.638
−	+	−	−	−	0.784	0.618
+	+	−	−	−	0.855	0.695
−	−	+	−	−	0.823	0.672
−	−	−	+	−	0.837	0.698
−	−	+	+	−	0.907	0.777
+	+	+	+	−	0.932	0.825
−	−	−	−	+	0.946	0.841
+	+	−	−	+	0.984	0.956
−	−	+	+	+	0.987	0.964
+	+	+	+	+	0.991	0.976

[a]Specificity = TP/(TP + FP) at a sensitivity (SE = TP/(TP + FN)) of 0.90.
[b]The SVM classified on the basis of the acceptor and the donor consensus values.
Performances are indexed by ROC values and specificity. Each row is an SVM test. ROC values were measured in untouched sets of ~2200 real and ~2300 pseudo exons. The first five columns indicate the components used by SVM. US, upstream; DS, downstream; TP, true positive; FN, false negative; FP, false positive; SE, sensitivity; SP, specificity.

Figure 3.5: Result table from [Zhang et al., 2003]

This work uses SVMs as a means to gain biological knowledge about the nucleotide sequences. The tests in this work actually involve sequence representations corresponding to exons, and the patterns that might be significant for splicing are deduced from these. Moreover, the representation chosen for the sequences is quite limited. In this sense, we propose a more specific and exhaustive study of the prediction of splice sites using SVMs, by representing the sequences as occurrences of 1-, 2-, 3-, 4-, 5-, and 6-mers with an adaptable degree of position-independent information. We also present an intuitive visualisation of the decision function of the classifier, and therefore of the important features. The information gained by this means is compared with the output of three different feature selection methods. As a conclusion, we offer a very detailed and systematic study of the nucleotide composition in the neighbourhood of the splice site regions.

3.2.6 More complex prediction systems

An example for a more complex prediction system is the one proposed by Rajapakse and Ho [2005], which combines Markov chains and neural networks, as schematised in Figure 3.6. This model considers splice sites as a concatenation of three segments: upstream segment, signal segment, and downstream segment. These three segments are represented by a first-order Markov chain and two second-order Markov chains, respectively, and a neural network takes the output of the Markov chains as input. The total length of the sequence considered for the prediction is 189 nucleotides for the acceptor sites and 176 for the donor sites, both centred around the consensus dinucleotide.

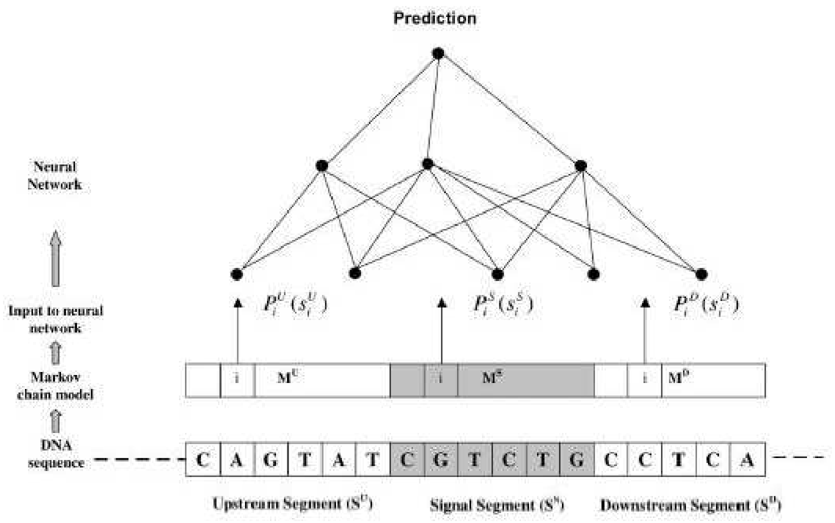


Figure 3.6: Splice site prediction model as suggested by Rajapakse and Ho [2005].

The approach introduced by Rajapakse and Ho [2005] showed to be more powerful than previous approaches, including GeneSplicer [Pertea et al., 2001], on the data sets were it was evaluated. GeneSplicer combines MDD models with Markov chains. Both systems require longer sequence windows (of more than 160 nucleotides), as they make use of the known differences between coding and non-coding sequences (e.g. codon usage), and are not directly intended to identify patterns biologically meaningful for splicing.

Due to the nature of the dependencies existing among base pairs in splicing sites, and the impossibility of obtaining a high-quality data set that is large enough for parameter estimation, accurate splice site prediction is still a complex problem, for which it remains necessary the development of new solutions or improvement of existing methods [Arita et al., 2002, Guigó et al., 2000]. In general, combinations of many different coding measures, and in particular the best methods available to capture each of these, should improve the solution for problems like the splice site prediction problem Salzberg [1997]. This concept motivates work in the direction of designing and improving existing methods for extracting information from biological sequences.

Chapter 4

Gene Prediction

4.1 Gene prediction based on sequence homology

AGenDA [Rinner and Morgenstern, 2002] is a gene prediction program that uses cross-species sequence comparison to identify functional regions. The concept is significantly different from other *homology-based* programs, such as PROCRUSTES [Gelfand et al., 1996] and GeneWise [Birney et al., 2004, Birney and Durbin, 1997], which work by looking for similarities between the query sequence and protein databases. The output of these programs is limited to sequences that present homology to some known protein sequences, and hence, they are incapable of identifying genes for which there is no *orthologous* (shared between species) or *paralogous* (resulting from a gene duplication event) gene recognised.

A common strategy for enhancing useful genetic information is through duplication, followed by variation, of functional segments of DNA. However, when the members of a gene family are examined, certain sequence regions appear to be more highly conserved than others. As a matter of fact, random mutation is followed by selection for those variations that alter the protein sequence providing phenotypic advantages, and selection against those variations that destruct function [Horrevoets et al., 1993]. As a consequence, functional regions are largely conserved during evolution. AGenDA makes use of this knowledge and predicts genes based on homologies found between two sequences belonging to related species. If the sequences correspond to syntenic regions, i.e., if both sequence and gene order are conserved between the two sequences, homologous sections can be useful hints for gene prediction.

AGenDA takes as input two evolutionary related genomic sequences, and the corresponding alignment annotation produced by DIALIGN; the selection of the sequences is crucial, as the gene prediction method is based on the homologies between them. The program basically clusters the alignments between the two sequences calculated by DIALIGN [Morgenstern, 1999] according to distance and frame conservation, and then searches and scores the boundaries of the resulting clusters for conserved splice signals. This is achieved in the following consecutive steps:

1. The homologous fragments between the two sequences identified by DIALIGN are clustered according to their relative position to each other in both sequences; by this means, homologous fragments that are at approximately the same distance smaller than a threshold t in one sequence and in the other, and with a gap that is compatible with the preservation of the reading frame, will be clustered together.

2. The next step is the identification of the dinucleotides characteristic for splice sites, i.e., AG for acceptor sites, and GT (GU in the pre-mRNA sequence) for donor sites, and trinucleotides

corresponding to the translation start, *ATG* (*AUG* in the pre-mRNA), and translation stop codons, *TAA*, *TGA* and *TAG* (*UAA*, *UGA* and *UAG* in the pre-mRNA). Each signal, represented by a dinucleotide/trinucleotide and a sequence window of fixed length around it, is then scored using the weight matrices proposed by Salzberg [1997]. Signals occurring outside the clusters are penalised proportionally to their distance to the closest cluster. The scores of the signals that are conserved among the two sequences are rewarded; a given signal in the first sequence is conserved if there is a signal of equivalent type (acceptor or donor site, start or stop translation codon) present at the same relative position with respect to the corresponding homologous cluster in the second sequence.

3. Finally, all possible combinations of acceptor sites/donor sites, acceptor sites/translation stops, translation starts/donor sites, and translation starts/translation stops contribute to the list of *candidate* exons; the alignment score of those exons whose sequences extend outside the borders of the homology clusters are penalised. The score for a given exon is

$$\begin{aligned} CE\,score &= start\,score \cdot start\,weight \\ &\quad + alignment\,score \cdot alignment\,weight \\ &\quad + end\,score \cdot end\,weight \end{aligned}$$

where *start score* corresponds to the score given to the start codon or acceptor splice site of the *candidate* exon, *end score* to the stop codon or donor splice site, and *alignment score* to the quality of the alignment, all updated according to the penalisation and rewarding schemes described above.

The output of AGenDA consists of a list of potential or *candidate* exons, which is used to construct complete gene models by a combinatorial optimisation approach detailed in the following section.

4.2 Gene assembly algorithm

The output of AGenDA includes the coordinates, frame, remainder, and general score of the candidate exons. Candidate exons are open reading frames delimited by signals; the coordinates indicate the first and last nucleotides of the sequence, the *frame* represents the reading frame with regard to the first nucleotide in the exon sequence, and the *remainder* is the number of nucleotides left after defining the last complete codon of the exon, given its frame. In order to predict genes, this information is subsequently used to find a biologically consistent *chain* of candidate exons (exon assembly), such that the sum of the scores of the candidate exons involved in the chain is maximal. This is done using a variation of a dynamic programming algorithm known as the *chaining problem* [Gusfield, 1997]. This algorithm is able to find an optimal chain of intervals in $O(r \log r)$ time, where r is the number intervals; the chain is optimal with regard to the sum of the scores associated with each participating interval. In the case of the gene assembly problem, the intervals correspond to the coordinates of the candidate exons, and the optimal chain is a gene structure with maximum total score.

The algorithm works as follows. Consider a set of r (possibly) overlapping intervals in a nucleotide sequence, where each interval j has some associated value $v(j)$; the value $v(j)$ represents some kind of likelihood that the interval is in fact an exon. The problem (see Figure 4.1) is to select a subset S of non-overlapping intervals, such that the sum of their associated values, $\sum_{j \in S} v(j)$, is maximal.

Let I be the list of all the $2r$ numbers representing the location of the endpoints (*coordinates*) of the candidate exons.

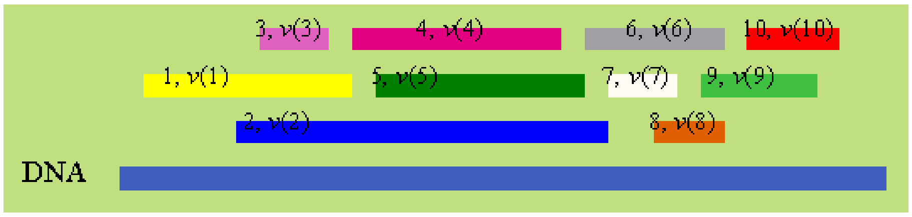


Figure 4.1: DNA sequence and $r = 10$ smaller intervals of the same sequence. Each interval j has associated a value v(j), and the problem is to choose a subset of non-overlapping intervals with maximal $\sum_j v(j)$.

```
Sort the numbers in $I$, and annotate each entry in $I$ with the name of the intervals it
is part of, and a comment of whether it is a left or a right ending. The sorting can
be accomplished efficiently through an implementation of the Quicksort algorithm (for
a description, see for example [Cormen et al., 1990]).
Set $max$ to zero.
For $i$ from 1 to $2r$ do
Begin
If $I(i)$ represents the left end of interval $j$, then set $V(j)$ to $v(j) + max$.
If $I(i)$ represents the right end of interval $j$, then set $max$ to the maximum of $max$
and $V(j)$.
End
```

The value of the variable *max* at the end of the algorithm is the value of the optimal solution, in other words, the score of the final exon assembly.

The algorithm considers matters that are particular to the gene assembly problem. From this point of view, acceptable chains must be biologically consistent in the following sense. Assuming that E_i is the candidate exon at position i in a possible chain, $fr(E_{i+1}) \in \{0, 1, 2\}$ is the frame of exon E_i, and $re(E_i) \in \{0, 1, 2\}$ is the remainder of exon E_i

- If E_i is on the forward strand and ends with a donor splice site, then E_{i+1} must be also on the forward strand and begin with an acceptor splice site, and $fr(E_{i+1}) + re(E_i) = 3$.

- If E_i is on the forward strand and ends with a stop codon, then E_{i+1} must begin with a start codon.

- The reading frame is preserved by all the exons constituting the chain.

The program allows obtaining either multiple gene models or single gene models, according to the user wish. In addition, the user can require the model to start with an exon that begins with a start codon (*First exon*), and/or to finish with an exon containing a stop codon (*Terminal exon*). There are also options to favour the inclusion of *First exons*, filter exons with a score lower than a threshold, and restrict the length of the chain (in terms of the number of exons included). The output produced by this program is the list of the exons that compose the predicted gene, with all the data available about them.

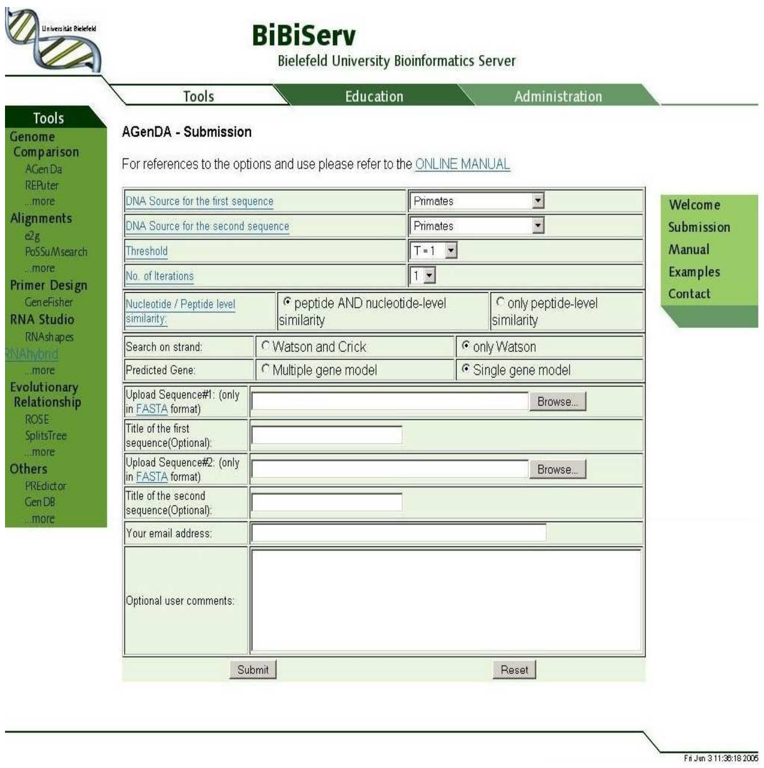
This algorithm is implemented in *C*, and due to its standard input and output formats could be employed (or easily modified to be employed) universally together with any gene prediction program.

4.3 Web server

To facilitate the access of end users to AGenDA, we have set up a web server and designed a web interface for AGenDA, enabling the same queries to be made via any Internet connection [Taher et al., 2004b, 2003]. The program is now publicly available at the Bielefeld University Bioinformatics Server (Bibiserv) and can be accessed through the URL `http://bibiserv.techfak.uni-bielefeld.de/agenda/`.

The input for the web server is a pair of potentially related genomic sequences, and information about the species they belong to. The input form can be observed in Figure 4.2. The web server automates a series of processing events to produce the input data that AGenDA requires. Such processing events consist on consecutively running the following programs

- RepeatMasker [Smit et al., 1996-2004]
- CHAOS [Brudno et al., 2003]
- DIALIGN [Morgenstern, 1999]
- AGenDA
- gene assembly algorithm



AGenDA - Submission

For references to the options and use please refer to the ONLINE MANUAL

DNA Source for the first sequence	Primates	
DNA Source for the second sequence	Primates	
Threshold	T = 1	
No. of Iterations	1	
Nucleotide / Peptide level similarity:	[✓ peptide AND nucleotide-level similarity]	only peptide-level similarity
Search on strand:	Watson and Crick	[✓ only Watson]
Predicted Gene:	Multiple gene model	[✓ Single gene model]
Upload Sequence#1: (only in FASTA format)		
Title of the first sequence(Optional):		
Upload Sequence#2: (only in FASTA format)		
Title of the second sequence(Optional):		
Your email address:		
Optional user comments:		

Figure 4.2: AGenDA web server submission form screenshot.

RepeatMasker is a program that masks interspersed repeats and low complexity DNA sequences, which would otherwise interfere with the sequence alignment. This is achieved by comparing the sequences using an efficient implementation of the Smith-Waterman-Gotoh algorithm. RepeatMasker annotates the repeats found in the query sequence, and also produces a copy of the query sequence where all the annotated repeats have been replaced by the character *N*. The fact that, on average, almost 50% of a human genomic DNA sequence will currently be masked by the program, illustrates why it is essential to mask the query sequence before applying a (local) alignment program.

The second program involved in the analysis is *CHAOS*, an algorithm for rapid identification of chains of local pair-wise sequence similarities. CHAOS calculates local alignments that are used as anchor points for *DIALIGN*. This is necessary because, in spite of being a very sensitive multiple-alignment tool, DIALIGN running times are usually rather long. The anchors limit DIALIGN search space, significantly speeding up the alignment without reducing its quality [Brudno et al., 2003]. Using the anchor points calculated by CHAOS, we calculate an alignment using DIALIGN.

The output of DIALIGN is a list of coordinates corresponding to the endpoints of the homologous regions found between the two sequences, associated with a measure of similarity. The alignment information, together with the sequences, constitutes the input for AGenDA. After running AGenDA and the gene assembly program, the output is returned to the user via e-mail; this e-mail contains the structure of the predicted gene, as a list of exon coordinates, together with the sequence of the predicted protein and information about the scoring involved; besides, the e-mail includes links to the lists of candidate exons and to a graphical representation of the alignments calculated by DIALIGN.

Figure 4.3 represents the sequence alignment calculated by CHAOS and DIALIGN, and the predicted gene model. The picture can be divided into four horizontal fields. In the top field, a bar graph represents the similarity between the two sequences; the colour of the bars indicates the type of similarity (blue for similarity at the nucleotide level, and red and orange bars for similarity at the peptide level on the positive and reverse strand), and the height of the bars represents the degree of similarity. The next field shows the gene model for the first input sequence (model 1), while the bottom field shows the gene model for the second input sequence (model 2). The third field is a scheme of the alignment between the two sequences.

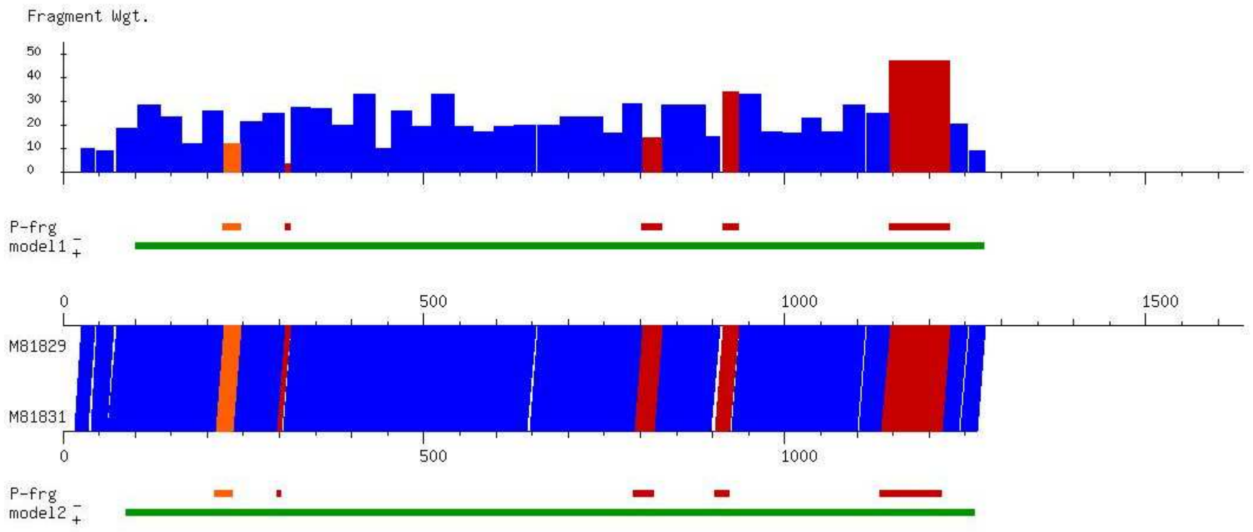


Figure 4.3: Graphical output of AGenDA. Blue bars between the input sequences represent sequence similarity at the nucleotide level, while red and orange bars represent similarity at the peptide level on the positive and reverse strands, respectively; the vertical bars indicate the degree of similarity of the local sequence homologies. The gene models predicted by AGenDA for the two sequences are represented with green bars.

4.4 Limitations of AGenDA

AGenDA requires sequences containing homologous regions, which are not always easy to obtain. In principle, this could be done using bidirectional BLAST searches on sequences for which there is experimental evidence of synteny; such information could be extracted from synteny maps like the one elaborated by the Sanger Institute. Figure 4.4 illustrates the synteny map obtained from the Sanger Institute website (`http://www.sanger.ac.uk/HGP/Chr22/Mouse/`) for human chromosome 22 and mouse; only the long arm (22q) of the chromosome is presented, as the short arm (22p) contains mainly tandem repeat structures and no protein coding genes.

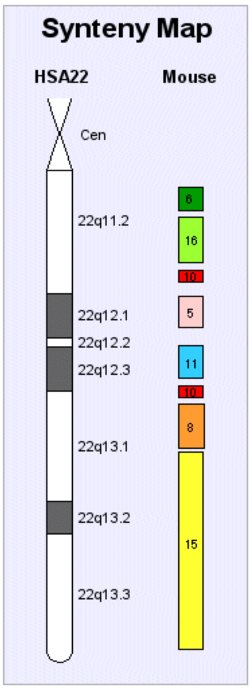


Figure 4.4: Synteny map for human chromosome 22 and mouse, obtained from the Sanger Institute website (`http://www.sanger.ac.uk/HGP/Chr22/Mouse/`). The numbers on the right indicate the matching chromosomes in mouse.

Nevertheless, as mentioned before, the input data is in the case of AGenDA a critical determinant of the results. Although cross-species sequence comparison can provide valuable information about the location of functional regions, and hence, of genes, it should be considered together with the evolutionary distance of the species involved (the sequences of very closely related species will be almost identical, while the sequences of far apart species will present almost no similarity). In the end, this makes the method applicable only to the sequences of species for which exhaustive synteny maps are available, which may imply that AGenDA's prediction is founded on information about the presence and placement of homologous genes (*synteny*). Of course, to some extend, this kind of assumption is common, and probably constitutes the strongest limitation of most gene prediction programs [Burge and Karlin, 1998].

Chapter 5

Splice Site Prediction

In this chapter, we present two approaches for predicting splice sites. In the first approach, we show how to use Support Vector Machines (SVMs) to extract features that distinguish between real splice site sequences and pseudo splice site sequences (that is, sequences containing the conserved AG or GT which do not correspond to known splice sites). The second approach is based on position-specific weight matrices (WMMs); with regard to this, we propose a method to overcome problems related to the amount of data available when estimating position-specific weight matrices, and apply it to the prediction of splice sites.

5.1 Splice site prediction using SVMs

5.1.1 Introduction

Previous work about the application of SVMs to the splice site prediction problem is limited regarding the representation of the sequences [Baets et al., 2002, Zhang et al., 2003, Saeys et al., 2004]. This is the motivation for the present more comprehensive and systematic study of the prediction of splice sites using SVMs. We represent the splice site sequences by occurrences of 1-, 2-, 3-, 4-, 5-, and 6-mers with different degrees of positional information, and study their influence on the performance of the SVM. After confirming that our SVM model possess a good prediction power, we use the decision function of the classifier to identify the features that are relevant for the classification, and hence the distinction between real splice sites and pseudo splice sites. Our model shows a strong capability for capturing motifs of biological meaning.

In the next two sections, we preliminarily evaluate different sequence representations for the prediction of splice sites; the conclusions result in the kernel presented in section 5.2.

5.1.2 Exploratory experiments with SVMs

Introduction

SVMs work by separating examples into two or more *classes*. This separation is achieved by means of a hyperplane, that results from solving a quadratic optimisation problem. For this reason, the most demanding task for the generation of an appropriate model consists on providing a vector representation that optimises the separation of the elements we are interested in, as this determines the information input to the learning system, and thereby, the model. With the aim of proposing a suitable representation for splice site sequences, we first essayed some basic representations and studied the corresponding variation in the performance of SVMs.

Method

Datasets

The experiments presented here were performed on the HS3D sequences.

Sequence representation

The instances or examples that serve as input to the SVM are nucleotide sequences of fixed length that contain the conserved dinucleotide AG (acceptor splice sites) or the conserved dinucleotide GT (or GU in the pre-mRNA, for donor splice sites) in an appropriate mathematical representation. The examples are labelled +1, if they represent a (true) splice site, or −1, if they represent a false (pseudo) splice site.

Our sequences are represented in terms of the substrings or k-mers of fixed length k that they contain. The nucleotide sequence can be thought of as a string of characters from the nucleotide alphabet $\Sigma = \{A, C, G, T\}$ (for RNA, the T is replaced by U). In this context, k-mers are lists or ordered sets of elements $(a_1, a_2, \ldots, a_k)$ where $a_i \in \Sigma$ for all $i = 1, 2, \ldots, k$. A sequence x can be represented by a vector whose components indicate the presence or absence of certain k-mers associated or not with a position in the sequence; the components of the vector are also known as *features*. We consider two types of feature vectors.

- *Vectors with position-dependent features*

  These vectors consist of position-dependent information exclusively. Their components represent the presence or absence of k-mers at specific positions in a window $[l, r]$, where $l, r \in \mathbb{Z}$, $l < r$, and 0 corresponds to the occurrence of the first character of the conserved dinucleotides AG or GT. Let $q_i \in \Sigma^k$ be a particular k-mer, so that $\Sigma^k = \{q_1, q_2, \ldots, q_{4^k}\}$, and let $S_{q_i} = \{p_1^{q_i}, p_2^{q_i}, \ldots\}$ be the set of positions where the first character of k-mer q_i occurs in x. Then, for $i = 1, 2, \ldots, 4^k$, we define the functions

$$\nu_{q_i} = \begin{pmatrix} \nu_{q_i}(l) \\ \nu_{q_i}(l+1) \\ \vdots \\ \nu_{q_i}(r-k+1) \end{pmatrix} \tag{5.1}$$

  where

$$\nu_{q_i}(t) = \begin{cases} 1, & \text{if } t \in S_{q_i} \\ 0, & \text{otherwise} \end{cases} \tag{5.2}$$

  for $l \leq t \leq r$. This kind of vector is also known in computer sciences as *characteristic vector*.

  Each sequence x can be mapped into a vector of the form

$$\phi(x) = \begin{pmatrix} \nu_{q_1} \\ \nu_{q_2} \\ \vdots \\ \nu_{q_{4^k}} \end{pmatrix}. \tag{5.3}$$

As exactly one k-mer occurs at each position in the sequence, these vectors include $(4^k - 1) \cdot [(r - l + 1) - k + 1]$ components with value zero, and in order to optimise the calculations and the memory allocation, they can be made *sparse*. This is done by explicitly representing merely the non-zero features, as pairs of feature indices and values.

For example, if we define $q_1 = A$, $q_2 = C$, $q_3 = G$, $q_4 = T$, the 5-nucleotide-long sequence $AACGT$ is represented in term of its constitutive monomers (1-mers) as

$$\phi(x) = [1:1, 2:1, 8:1, 14:1, 20:1]^T . \tag{5.4}$$

This representation indicates the presence of features with indices 1, 2, 8, 14 and 20, and the absence of the remaining 15 features.

The sparse representation in terms of trimers (3-mers) of the same sequence is a vector of 192 components, three for each of the 64 possible substrings of length three that can be built using the nucleotide alphabet. If $q_1 = AAA$, $q_2 = AAC$, ..., $q_{64} = TTT$ (i.e., the trimers are alphabetically sorted in ascending order), the sequence $AACGT$ is represented in terms of its constitutive trimers as

$$\phi(x) = [4:1, 8:1, 30:1]^T . \tag{5.5}$$

This simple representation provides the learning system with information about the patterns that the sequence contains and their positions. Position-dependent k-mers should provide an appropriate representation for conserved sequences, such as short regions comprising the AG in the case of acceptor sites, and the GT for donor sites.

- *Vectors with position-independent features*

  These vectors consist exclusively of position-independent information; a similar approach has been already used, for example, for protein classification [Leslie et al., 2002]. Their components represent the presence (1) or absence (0) of k-mers in a window $[l, r]$, where $l, r \in \mathbb{Z}$, $l < r$, and 0 corresponds to the first character of the conserved dinucleotide AG or GT; however, such components do not include any information referring to the position of the k-mers in the window.

  The sequences are represented by vectors whose components contain the number of occurrences of each possible k-mer in Σ^k, with $\Sigma = \{A, C, G, T\}$. Consequently, these vectors contain $|\Sigma^k| = 4^k$ components, one for each possible k-mer of length k that can be constructed with the characters of alphabet Σ. Let $q_i \in \Sigma^k$ be a particular k-mer, so that $\Sigma^k = \{q_1, q_2, \ldots, q_{4^k}\}$, and let $S_{q_i} = \{p_1^{q_i}, p_2^{q_i}, \ldots\}$ be the set of positions where k-mer q_i occurs in x. Each sequence x can be mapped into a vector of the form

$$\phi(x) = \begin{pmatrix} \nu_{q_1} \\ \nu_{q_2} \\ \vdots \\ \nu_{q_{4^k}} \end{pmatrix} \tag{5.6}$$

  where

$$\nu_{q_i} = |S_{q_i}|. \tag{5.7}$$

  For example, a vector $\phi(x)$ representing occurrences of trimers (3-mers) has 64 components. If

$q_1 = AAA$, $q_2 = AAC$, …, $q_{64} = TTT$, the sparse representation for the sequence $AACGT$ in terms of trimers is

$$x = [2:1, 7:1, 28:1]^T \,. \tag{5.8}$$

Experimental setup

We separated the splice site prediction problem into two independent sub-problems: donor site prediction and acceptor site prediction. Each of these two sub-problems can be formally stated as a two-class classification problem: {true donor site, false donor site} and {true acceptor site, false acceptor site}.

The experiments were performed with a 3-fold cross-validation scheme; the parameter selection was accomplished through an internal 5-fold cross-validation scheme. The SVM hyperparameter C was optimised within the set $\{0, 0.01, 0.02, \ldots, 0.2, 0.25, 0.3, 0.35, \ldots, 1\}$. We evaluated two different ratios of negative to positive examples (1 and 10), and two windows over the sequence, $[-50, +50]$ and $[-25, +25]$.

Classification criteria

The performance of the classification methods is evaluated in terms of the area under the ROC curve (AUC) and the classification rate (CR).

Results

Figure 5.1 summarises the performance of SVMs on acceptor sites; the axes in these plots are drawn on the same scale to facilitate the comparison of the curves. Similarly, Figure 5.2 summarises the performance of SVMs on donor sites.

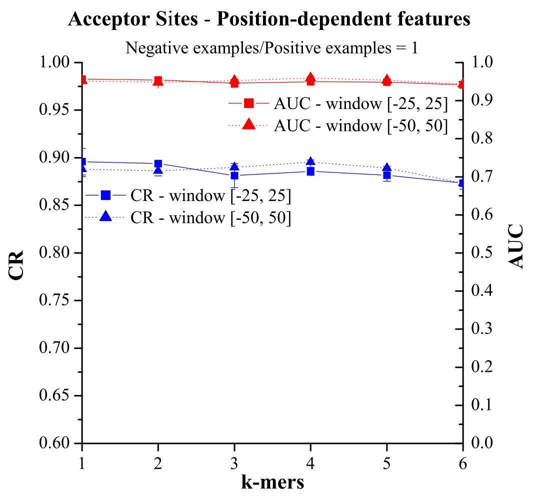
Discussion and conclusions

For vectors composed by position-dependent features, there are no strong differences between the AUC values obtained with datasets including positive and negative examples in equal amounts and the AUC values obtained with datasets containing ten times more negative than positive examples; however, the classification rates do differ. In general, vectors including longer position-dependent k-mers result in better performances than shorter k-mers.

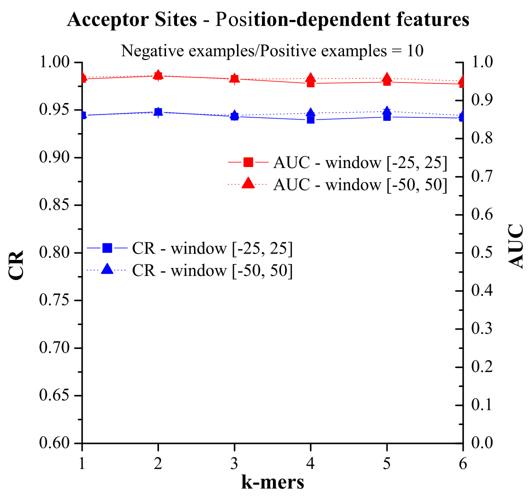
Regarding the classification of splice sites with vectors composed by position-independent features, there are differences between the AUC values and the classification rates. Nevertheless, the curve share approximately the same tendency; as expected, the use of short position-independent k-mers (monomers and dimers) does not result in a good classification; the classification improves for longer k-mers, but the improvement seems to have a limit with vectors composed by hexamers. The shorter window outperforms the longer window, except for vectors composed by short k-mers and datasets containing ten times more negative than positive examples.

The best classification for acceptor sites is achieved using position-dependent monomers and dimers, but position-independent 4-mers and 5-mer exhibit a good classification power as well. Regarding donor sites, the best model corresponds to vectors composed of position-dependent long k-mers, although all values of k produce comparably good results; indeed, the consensus region of donor sites is known to be longer than that of acceptor sites; position-independent long k-mers also result in good CR and AUC values.

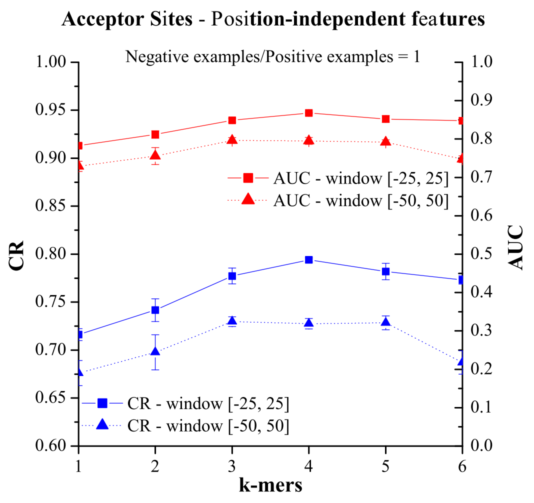
In general, the values of the classification rate (CR) and the area under the ROC (AUC) curve correlate well. Nevertheless, when the datasets involved contain positive and negative examples



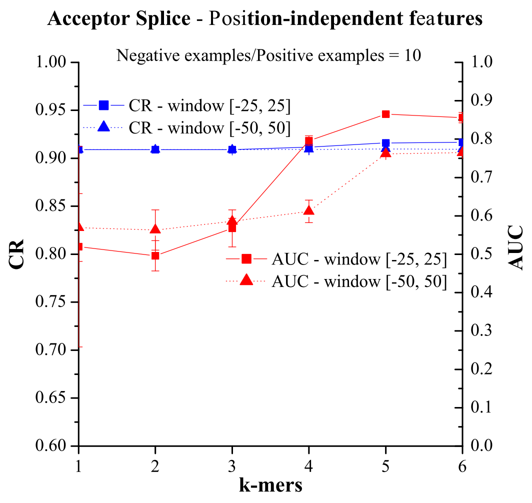
(a)



(b)



(c)

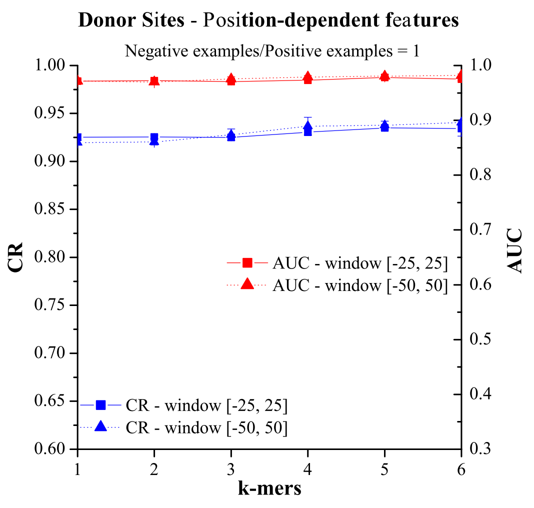


(d)

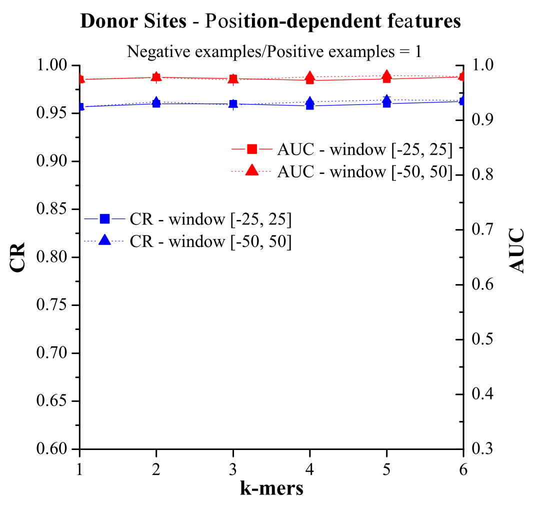
Figure 5.1: Acceptor site classification using SVMs. Preliminary tests. (a) and (b) correspond to vectors composed by position-dependent features; (c) and (d) correspond to vectors composed by position-independent features. The ratios between negative and positive examples are 1 ((a) and (c)), and 10 ((b) and (d)).

in a ratio different than one, the classification rate is dominated by the most numerous group, and the variation in the CR values is small, indicating that the CR is not the optimal classification criteria. Long windows provide a slight advantage when considering vectors composed of long position-dependent k-mers and short position-independent k-mers (the latter is probably related to a better capture of the sequence composition), and a disadvantage when considering vectors composed by long position-independent k-mers; otherwise, the choice of the window does not show to be critical. Although the CR and AUC values depend on the ratio of negative to positive examples, the curves show comparable tendencies; the increase of the ratio between negative and positive examples is limited by the computation time.

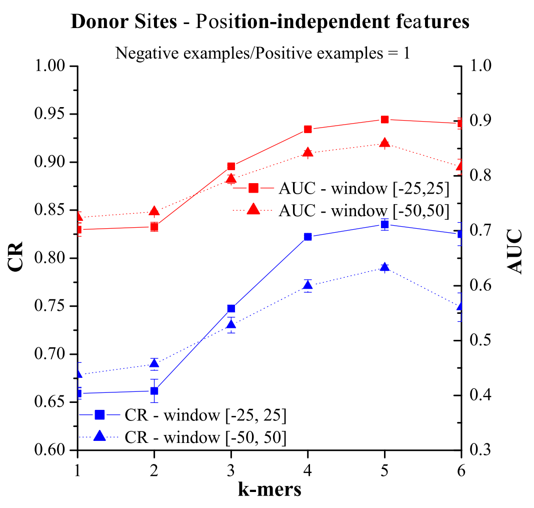
These results motivate our further approach to the splice site prediction problem using SVMs.



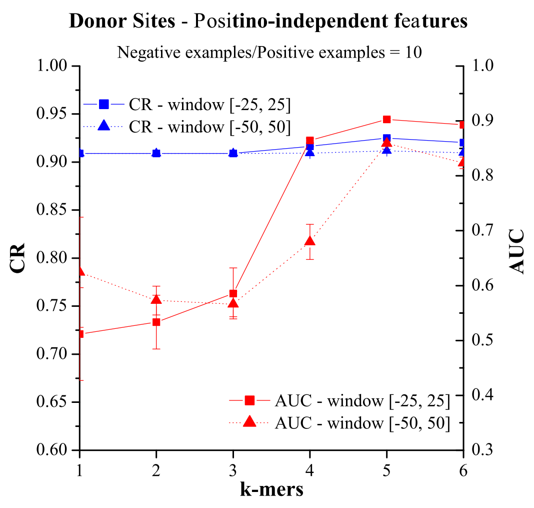
(a)



(b)



(c)



(d)

Figure 5.2: Donor site classification using SVMs. Preliminary tests. (a) and (b) correspond to vectors composed by position-dependent features; (c) and (d) correspond to vectors composed by position-independent features. The ratios between negative and positive examples are 1 ((a) and (c)), and 10 ((b) and (d)).

Our aim is to achieve an appropriate integration of position-dependent and position-independent information, and implement this into a sequence representation that allows deeper insight into the nucleotide patterns that are responsible for good prediction rates.

5.1.3 Exploratory experiments with PSVMs

Introduction

In order to evaluate the possibility of applying SVMs to the splice site prediction problem with the aim of extracting information about the patterns that are relevant for splicing, we further tested the influence of different feature representations on the performance of PSVMs. We chose PSVMs over PSVMs because of the advantages regarding implementation and speed of the former systems. The feature representations examined in these tests include features containing a variable and controllable degree of position-invariance, achieved by means of Gaussian smoothing. The results obtained constitute a general overview of different kinds of feature vectors, and direct the course of further research.

Method

Datasets

The experiments presented here were performed on the HS3D sequences. Only the signals labelled as *proximal* were employed as negative examples; as mentioned before, the dataset contains 2880 acceptor sites and 2796 donor sites, as well as 21,222 *proximal* pseudo acceptor sites and 17,431 *proximal* pseudo donor sites.

Gaussian Smoothing

Smoothing is a procedure commonly applied in image and audio editing to reduce noise. The basic process of smoothing consists in generating for each data point a new value that is some function of the original value at that point and the surrounding data points. In the case of Gaussian smoothing, this function is the Gaussian function, which has the form

$$g(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{5.9}$$

where μ is the mean and σ is the standard deviation of the distribution. The Gaussian distribution is illustrated in Figure 5.3 for $\mu = 0$ and different values of σ.

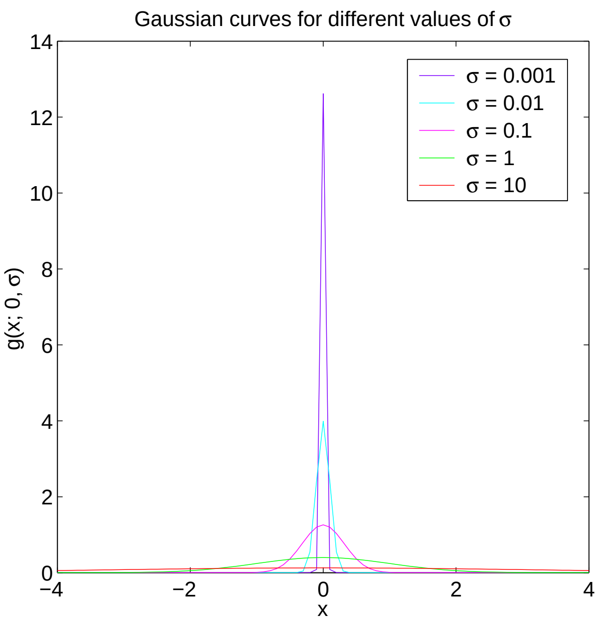


Figure 5.3: Gaussian distribution with $\mu = 0$ and different values of σ

Given a series of n points (signals) in $\mathbb{R}$, $t_1, t_2, \ldots$ with values (intensities) $\nu(t_1), \nu(t_2), \ldots$, the new value of a point $\mu \in T = \{t_1, t_2, \ldots\}$ will be given by

$$\nu'(\mu) = \sum_{t_i \in T} g(t_i; \mu, \sigma) \cdot \nu(t_i). \tag{5.10}$$

The procedure can be easily generalised to data in more dimensions. The parameter σ defines the shape of the Gaussian curve, and then determines the smoothing. Figure 5.4 shows the consequences of applying Gaussian smoothing to 11 randomly generated binary points.

In our case, the data are discrete positional occurrences of k-mers in the nucleotide sequences, and the effect of the Gaussian smoothing is to blur the positions where the k-mers occur. If a k-mer occurs at position μ, the degree of smoothing is determined by the standard deviation σ of the Gaussian curve. Larger values of σ increase the blurring; the limit cases $\sigma = 0$ and $\sigma \to \infty$ correspond to position-dependent information and to no position-dependent information at all, respectively. Figure 5.3 shows the values of σ that we used in our experiments, and the corresponding positions in the sequence where the Gaussian distribution is in practice different from 0, assuming that the smoothed k-mer occurs at position 0 ($\mu = 0$).

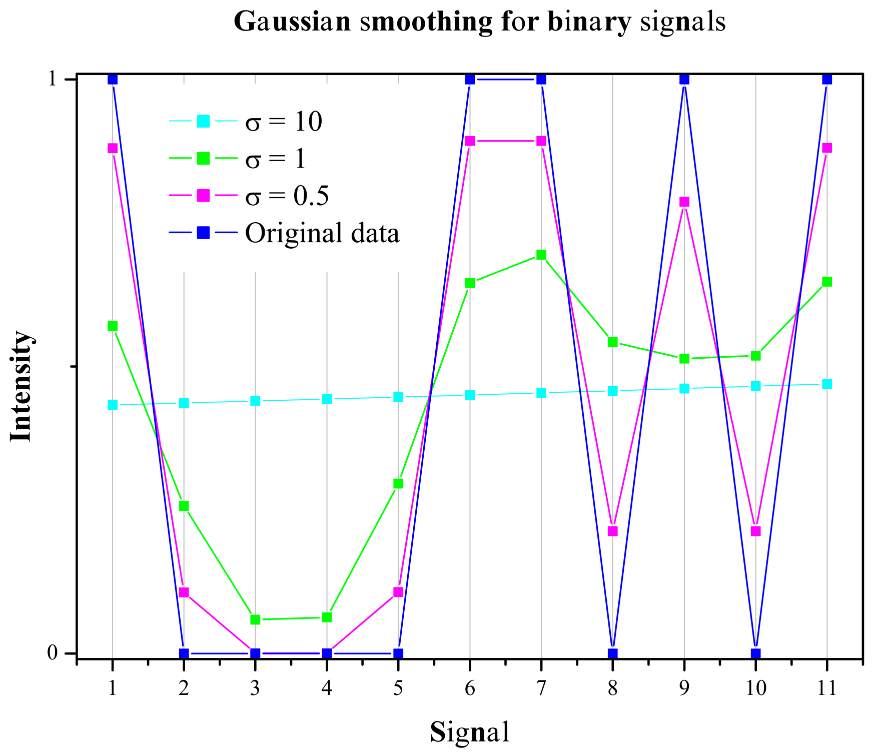


Figure 5.4: Application of Gaussian smoothing on binary data.

Sequence representation

The following experiments were performed using a window of 51 nucleotides centred at the first nucleotide of the splice site conserved dinucleotides, i.e., $[-25, +25]$.

The vectors were composed by

- position-dependent features
- smoothed k-mers
- position-independent features

- combinations of different features

The position-dependent features chosen were the positions of the occurrences of k-mers in the sequence, for small integers $k \in \{1, 2, 3\}$, while position-independent features are merely the occurrences of k-mers (with $k \in 1, 2, 3, 4, 5$), without any information about their location in the sequence. Vectors with position-dependent features contain $4^k (51 - k + 1)$ components, while vectors with position-independent features contain only 4^k components.

We also tested a representation in which the positions where the k-mers occur were blurred using a Gaussian smoothing operator, a commonly employed strategy for noise reduction in image processing. The objective here is to obtain features with a controllable amount of positional information, and in between position-dependent and position-independent features. The parameter controlling the amount of positional information corresponds to the standard deviation σ of the Gaussian distribution. Given a k-mer $q \in \{A, C, G, T\}^k$ occurring at positions $P = \{p_1, p_2, \ldots\}$ in the nucleotide sequence, we define the value of the feature corresponding to k-mer q at position $\mu \in P$ as

$$f^q_\mu = \frac{1}{c} \sum_{i \in P} e^{-\frac{(\mu - i)^2}{2\sigma^2}}, \tag{5.11}$$

where σ is the parameter controlling the degree of smoothing, and c is an appropriate normalisation constant. The vectors containing smoothed k-mer occurrences are composed of $4^k (51 - k + 1)$ features, with values of $k \in \{1, 2, 3\}$.

Finally, the vectors combining different kinds of features are composed of vectors with position-dependent (including also the smoothing) and position-independent features, normalised so that their L_2-norms are 1, and then heaped together. For these vectors, we used position-dependent monomers, position-independent pentamers and Gaussian-smoothed trimers.

Experimental setup

We separated the splice site prediction problem into the prediction of donor sites and the prediction of acceptor sites. The experiments were performed following a 3-fold 5-fold cross-validation scheme; the five internal cross-validation loops were used for adjusting the parameters C and σ; the three external cross-validation loops were used to test the model with optimal parameters on unseen data. Parameter C was selected from the set $\{10 \cdot 0.9^x \mid x \in \mathbb{N}_0 \textit{ and } x \leq 100\}$ and parameter σ from $\{0.025, 0.5, 0.75, 1.0, 1.5, 2.0\}$. The results presented here are averages over the three external cross-validation loops.

The sequence window employed in this experiments was fixed and equal to $[-25, +25]$, where 0 is the position of the first character of the dinucleotide AG (acceptor sites) or GT (donor sites). The datasets employed for these tests consisted of 2880 positive examples and 21,222 negative examples for the acceptor site prediction tests, and 2796 positive examples and 17,431 negative examples in the case of donor site prediction. For the realisation of PSVMs we used our MATLAB® implementation.

Reference method for comparison

We compared the performance of PSVMs with the performance of position-specific weight matrices (WMMs) of order 0, 1, and 2 on exactly the same data partitions. The length of the sequence window was an additional parameter of the position-specific weight matrix model; we selected the sequence window from the set of windows centred at position 0 (corresponding to the occurrence of the first character of the dinucleotides that characterise splice sites, AG and GT) with lengths 11, 21, 31, 41, and 51-k, where k is the order of the WMM. We estimated two models, one for true splice sites (positive class) and one for pseudo splice sites (negative class), and classify the test examples according to their probability of belonging to either of the classes. The position-specific weight matrices were also implemented in MATLAB®.

Classification criteria

The performance of the classification methods was evaluated in terms of the area under the ROC curve (AUC) and the classification rate (CR).

Results

This section summarises the results obtained for the classification of acceptor and donor splice sites using PSVMs and the four different sequence representations described above.

Acceptor Sites

- *Vectors with position-dependent features*

Table 5.1 and Figure 5.5 present the performance of SVMs for acceptor splice site sequences represented by position-dependent k-mers. Table 5.2 shows the performance of position-specific weight matrices on the same data, as a reference.

k	C	CR	AUC
1	0.274 ± 0.003	0.861 ± 0.003	0.955 ± 0.001
2	0.396 ± 0.004	0.864 ± 0.004	0.960 ± 0.002
3	0.189 ± 0.002	0.861 ± 0.002	0.961 ± 0.002

Table 5.1: Acceptor site classification using PSVM. Performance for vectors with position-dependent features.

order	window length	CR	AUC
1	49 ± 0	0.896 ± 0.002	0.959 ± 0.001

Table 5.2: Acceptor site classification using positional-weight matrix performance (reference).

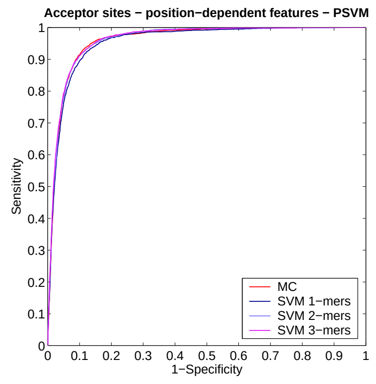


Figure 5.5: Acceptor site classification using PSVM. Performance for vectors with position-dependent features.

- *Vectors with smoothed k-mer occurrences*

Table 5.3 and Figure 5.6 present the performance of SVMs for acceptor splice site sequences represented by k-mers whose exact locations have been smoothed using a Gaussian operator. Table 5.4 shows the performance of position-specific weight matrices on the same data, as a reference.

k	C	σ	CR	AUC
1	0.25 ± 0.05	0.025 ± 0.000	0.863 ± 0.005	0.956 ± 0.000
2	0.28 ± 0.00	0.500 ± 0.000	0.8701 ± 0.0009	0.960 ± 0.001
3	0.07 ± 0.00	1.000 ± 0.000	0.883 ± 0.008	0.964 ± 0.001

Table 5.3: Acceptor site classification using PSVM. Performance for vectors with smoothed k-mer occurrences.

order	window length	CR	AUC
1	49 ± 0	0.896 ± 0.004	0.959 ± 0.002

Table 5.4: Acceptor site classification using positional-weight matrix performance (reference).

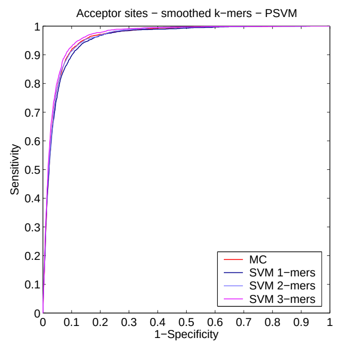


Figure 5.6: Acceptor site classification using PSVM. Performance for vectors with smoothed k-mer occurrences.

- *Vectors with position-independent features*

Table 5.5 and Figure 5.7 present the performance of SVMs for acceptor splice site sequences represented by position-independent k-mers. Table 5.6 shows the performance of position-specific weight matrices on the same data, as a reference.

k	C	CR	AUC
1	0.002 ± 0.002	0.7167± 0.0005	0.81 ± 0.01
2	0.0003 ± 0.0000	0.734± 0.007	0.817± 0.003
3	0.007 ± 0.003	0.76± 0.01	0.844± 0.005
4	0.010 ± 0.001	0.781± 0.007	0.86± 0.01
5	0.0475 ± 0.0000	0.73± 0.02	0.854± 0.007
6	0.008 ± 0.001	0.783± 0.003	0.866± 0.004

Table 5.5: Acceptor site classification using PSVM. Performance for vectors with position-independent features.

orde	window length	CR	AUC
1	49 ± 0	0.896 ± 0.004	0.959 ± 0.003

Table 5.6: Acceptor site classification using positional-weight matrix performance (reference).

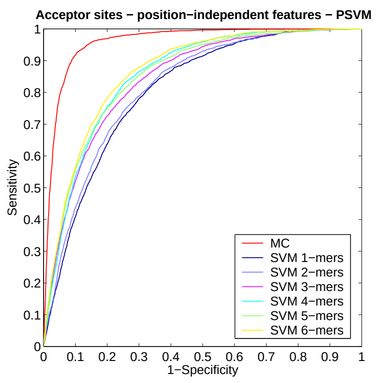


Figure 5.7: Acceptor site classification using PSVM. Performance for vectors with position-independent features.

- *Vectors with combinations of features*

Table 5.7 and Figure 5.8 present the performance of SVMs for acceptor splice site sequences represented by position-dependent monomers, trimer occurrences smoothed with a Gaussian operator, and position-independent pentamers. Table 5.8 shows the performance of position-specific weight matrices on the same data, as a reference.

C	σ	CR	AUC
0.04748 ± 0.0000	0.025 ± 0.000	0.858 ± 0.002	0.951 ± 0.003

Table 5.7: Acceptor site classification using PSVM. Performance for vectors composed by position-dependent 1-mers, smoothed 3-mers, and position-independent 5-mers.

order	window length	CR	AUC
1	49 ± 0	0.895 ± 0.007	0.959 ± 0.004

Table 5.8: Acceptor site classification using positional-weight matrix performance (reference).

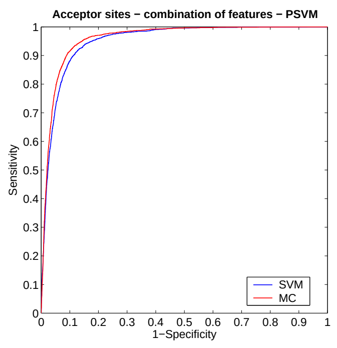


Figure 5.8: Acceptor site classification using PSVM. Performance for vectors composed by combinations of different features.

- *Vectors with position-dependent features*

Table 5.9 and Figure 5.9 present the performance of SVMs for donor splice site sequences represented by position-dependent k-mers. Table 5.10 shows the performance of position-specific weight matrices on the same data, as a reference.

k	C	CR	AUC
1	0.15 ± 0.03	0.890 ± 0.002	0.967 ± 0.003
2	0.40 ± 0.02	0.913 ± 0.007	0.973 ± 0.002
3	0.143 ± 0.008	0.916 ± 0.006	0.973 ± 0.003

Table 5.9: Donor site classification using PSVM. Performance for vectors with position-dependent features.

order	window length	CR	AUC
0	17 ± 6	0.875 ± 0.007	0.957 ± 0.003

Table 5.10: Donor site classification using positional-weight matrix performance (reference).

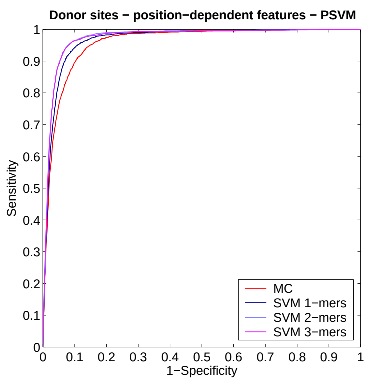


Figure 5.9: Donor site classification using PSVM. Performance for vectors with position-dependent features.

- *Vectors with smoothed k-mer occurrences*

Table 5.11 and Figure 5.10 present the performance of SVMs for donor site sequences represented by Gaussian-smoothed occurrences of k-mers. Table 5.12 shows the performance of position-specific weight matrices on the same data, as a reference.

k	C	σ	CR	AUC
1	0.158 ± 0.03	0.025 ± 0.000	0.889 ± 0.001	0.966 ± 0.003
2	0.1977 ± 0.0000	0.5 ± 0.0	0.913 ± 0.002	0.973 ± 0.002
3	0.0969 ± 0.0000	0.5± 0.0	0.918 ± 0.005	0.973 ± 0.002

Table 5.11: Donor site classification using PSVM. Performance for vectors with smoothed k-mer occurrences.

order	window length	CR	AUC
0	16 ± 5	0.875 ± 0.004	0.957 ± 0.003

Table 5.12: Donor site classification using positional-weight matrix performance (reference).

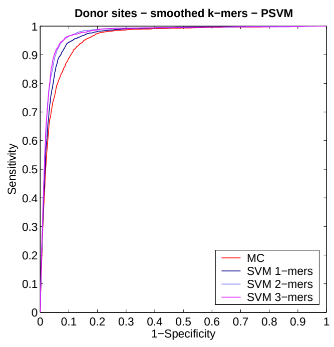


Figure 5.10: Donor site classification using PSVM. Performance for vectors with smoothed k-mer occurrences.

- *Vectors with position-independent features*

Table 5.13 and Figure 5.11 present the performance of SVMs for donor splice site sequences represented by position-independent k-mers. Table 5.14 shows the performance of position-specific weight matrices on the same data, as a reference.

k	C	CR	AUC
1	0.002 ± 0.0002	0.564 ± 0.007	0.64 ± 0.01
2	0.015 ± 0.003	0.613 ± 0.02	0.65 ± 0.01
3	0.0016 ± 0.0004	0.664 ± 0.03	0.771 ± 0.007
4	0.0059 ± 0.0007	0.76 ± 0.01	0.830 ± 0.004
5	0.010 ± 0.001	0.775 ± 0.003	0.854 ± 0.006
6	0.0078 ± 0.0008	0.800 ± 0.005	0.858 ± 0.001

Table 5.13: Donor site classification using PSVM. Performance for vectors with position-independent features.

order	window length	CR	AUC
0	17 ± 6	0.875 ± 0.004	0.958 ± 0.003

Table 5.14: Donor site classification using positional-weight matrix performance (reference).

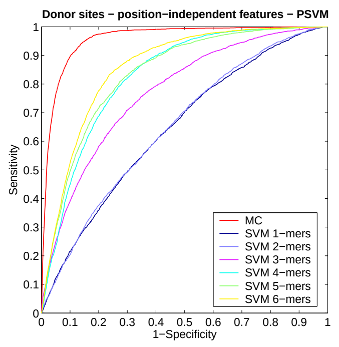


Figure 5.11: Donor site classification using PSVM. Performance for vectors with position-independent features.

- *Vectors with combinations of features*

Table 5.15 and Figure 5.12 present the performance of SVMs for donor splice site sequences represented by vectors composed by position-dependent 1-mers, smoothed 3-mers, and position-independent 5-mers. Table 5.16 shows the performance of position-specific weight matrices on the same data, as a reference.

C	σ	CR	AUC
0.0475 ± 0.0000	0.025 ± 0.000	0.899 ± 0.003	0.968 ± 0.004

Table 5.15: Donor site classification using PSVM. Performance for vectors composed by position-dependent 1-mers, smoothed 3-mers, and position-independent 5-mers.

order	window length	CR	AUC
0	17 ± 6	0.875 ± 0.005	0.958 ± 0.004

Table 5.16: Donor site classification using positional-weight matrix performance (reference).

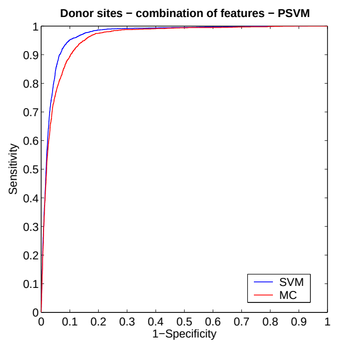


Figure 5.12: Donor site classification using PSVM. Performance for vectors composed by combinations of different features.

Discussion

Acceptor Sites

Figure 5.13 summarises the performance of PSVMs on acceptor sites. The most relevant findings are described in the following text.

- *Vectors with position-dependent features*

  Vectors composed of trimers perform best both in terms of the CR and the AUC, but monomers and dimers also perform comparably well. The performance of the SVM is comparable to the performance achieved by the position-specific weight matrix of order 1 (position-dependent occurrences of dimers).

- *Vectors with smoothed k-mers*

  The value of σ that produces the best model in terms of the CR and the AUC increases with k, the length of the k-mers considered. In the case of monomers, the best classification is achieved with very small values of σ ($\sigma = 0.025$), indicating the importance of position-dependent information. In the case of dimers, the best discrimination is achieved with $\sigma = 0.5$, which implies the existence of patterns with some positional variability. The best discrimination is achieved with trimers and $\sigma = 1$, indicating the relevance of patterns occurring within a range of positions. The best SVM outperforms the best weight matrix model, although not in terms of the classification rate.

- *Vectors with position-independent features*

  The AUC produced by the PSVM improves with the increase of k, the length of the k-mers considered. The vectors with the highest discrimination power are those composed by 6-mers, 4-mers and 5-mers; the vectors composed by 4-mers produce slightly higher classification rates than those composed by 5-mers. The improvement in the PSVM performance with the increase of k slows down for larger values of k, suggesting a limit. As expected, position-specific weight matrices clearly outperform the PSVMs.

- *Vectors with combinations of features*

  For equal AUC values, position-specific weight matrices produce better CR values. The best PSVM corresponds to a small value of σ (0.025), which suggests that the integration of features is not appropriate. Indeed, the smoothing affects only the trimers included in the vectors, which have been shown to be more effective with relatively large values of σ.

In general, regarding the CR, position-specific weight matrices perform better than PSVMs.

Donor Sites

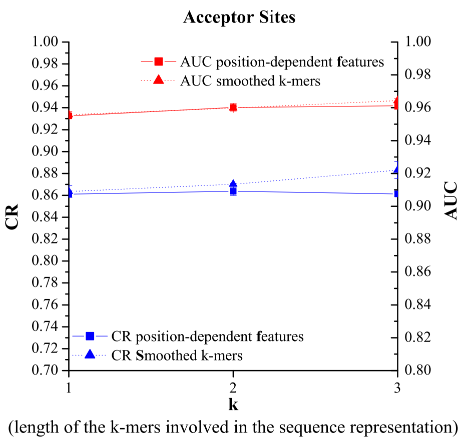
Figure 5.14 summarises the performance of PSVMs on donor sites. The most important results are reported below.

- *Vectors with position-dependent features*

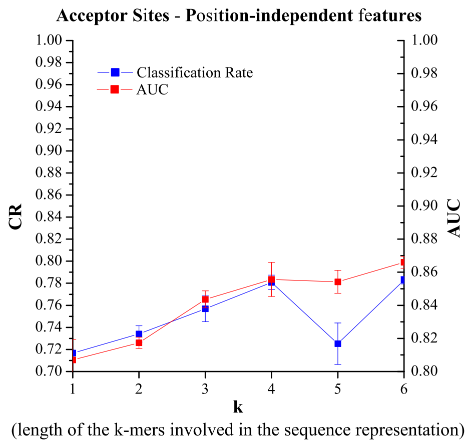
  Vectors composed of trimers perform best both in terms of the CR and AUC. The performance of the SVM is significantly higher than the performance achieved by the positional-weight matrices of order 0 (position-dependent occurrences of monomers).

- *Vectors with smoothed k-mers*

  In the case of monomers, the best classification is achieved with very small values of σ, with an optimal value of $\sigma = 0.025$, suggesting that donor sites are characterised for the occurrence of



(a)



(b)

Figure 5.13: Acceptor site classification using PSVMs. Performance for vectors composed by position-dependent features (a), and position-independent features (b).

certain monomers at fixed positions. In the case of dimers and trimers, the best discrimination is achieved with $\sigma = 0.5$, implying the importance of certain dimers and trimers occurring with some positional variability. Vectors composed by trimers perform best, and outperform the best position-specific weight matrix model.

- *Vectors with position-independent features*

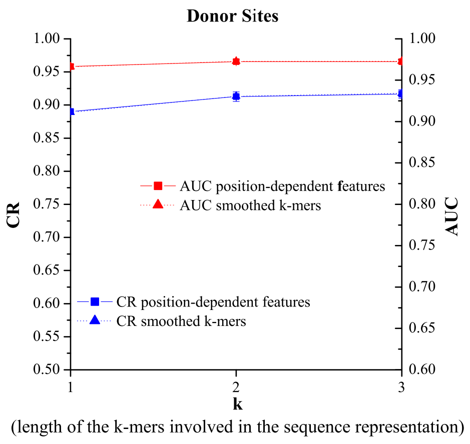
  The performance increases regularly with k, the length of the k-mers considered, and the best discrimination is achieved with vectors composed by hexamers. Position-specific weight matrix models distinctly outperform the PSVMs.

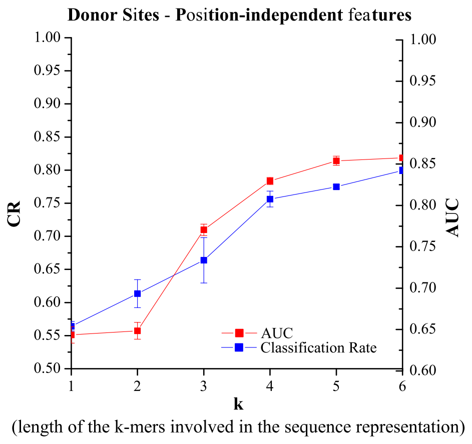
- *Vectors with combinations of features*

  The model does not outperform the best PSVM with non-combined feature vectors. The small value of σ (0.025) observed for the best classifier suggest that the feature combination is not optimal for solving the problem.

Conclusions

The results confirm that position-independent features contain valuable information that could contribute to improve the development of splice site prediction systems. The best overall model for acceptor sites corresponds to the vector representation containing smoothed trimers; although with a smaller value of σ, indicating the importance of position-dependent features, donor splice sites are also very well represented by smoothed dimers and trimers. These results motivate further study of vector representations composed by features associated with a variable degree of positional information.





(a) (b)

Figure 5.14: Donor site classification using PSVMs. Performance for position-dependent features (a), and position-independent features (b).

5.2 Identification of splicing-related patterns using the Oligo kernel

5.2.1 Introduction

Based on the evidence that indicates that position-independent information is potentially valuable for distinguishing real splice sites from pseudo splice sites, we propose a sequence representation that allows us to vary the amount of positional information we retain. For accomplishing this, we employ a process known as *smoothing*, whose primary purpose is to improve the continuity of the data and remove or diminish outlying points. Starting from a sequence represented by the positions where certain k-mers or *words* of length k occur, we smooth these positions, and thus relocate the k-mers to a range of positions, instead of preserving the original punctual information; by this simple procedure, we are able to add a flexible degree of uncertainty to the occurrence position of the k-mers. We perform this operation implicitly by means of a kernel function, which in principle allows sequence representations using occurrences of arbitrarily long k-mers.

The framework we propose comprehends the traditional representations used for nucleotide sequences, providing a more general sequence representation and facilitating the systematic study of the features involved in the prediction. In this sense, we also present a visualisation that supplies with an intuitive insight into the decision function of the SVM classifier, and hence into the features that are relevant for the classification of splice sites and for the underlying biological process.

We complement and support the analysis of the features involved in the classification by studying and comparing different feature selection methods based on the prediction outcome of SVMs. For this task we selected three methods: Weight Score filter [Furey et al., 2000], RFE-SVM algorithm [Guyon et al., 2002] and L_0-norm algorithm [Weston et al., 2003]. Although it has been shown that SVMs perform well in high-dimensional input spaces, by restricting the training and testing to a subset of the total features we intend to improve the classification performance, gain additional biological knowledge, and reduce the computational time.

5.2.2 Oligo kernel

Sequence representation (Primal form)

Let x be a nucleotide sequence in the window $[l, r]$, where $l, r \in \mathbb{Z}$, $l < r$, and 0 corresponds to the position where the first character of some biological signal occurs. Let $\Sigma = \{A, C, G, T\}$ be the DNA alphabet (or the RNA alphabet, if T is replaced by U). For $k \in \mathbb{N}$, let $q_i \in \Sigma^k$ be a particular k-mer, so that $\Sigma^k = \{q_1, q_2, \ldots, q_{4^k}\}$, and $S_{q_i} = \{p_1^{q_i}, p_2^{q_i}, \ldots\}$ be the set of positions where k-mer q_i occurs in x. In Equation 5.1 we have represented nucleotide sequences as vectors that contain information about the positions of the occurrences of all k-mers $q_i \in \Sigma^k$, according to

$$\phi(x) = \begin{pmatrix} \nu_{q_1}(l) \\ \nu_{q_1}(l+1) \\ \vdots \\ \nu_{q_1}(r-k+1) \\ \nu_{q_2}(l) \\ \nu_{q_2}(l+1) \\ \vdots \\ \nu_{q_{4^k}}(r-k+1) \end{pmatrix} \tag{5.12}$$

where each component or *feature* is

$$\nu_{q_i}(t) = \begin{cases} 1, & \text{if } t \in S_{q_i} \\ 0, & \text{otherwise} \end{cases} \tag{5.13}$$

for $l \leq t \leq r$.

Now, for each k-mer $q_i \in \Sigma^k$, with $i = 1, 2, \ldots, 4^k$, and each position μ in the sequence x, $l \leq \mu \leq r$, we are interested in assigning to the corresponding feature a value that takes into account all the occurrences of q_i in x; we achieve this through a Gaussian smoothing operator. For the vector representation of x, the component or *feature* that represents k-mer q_i at position μ takes the value

$$\nu'_{q_i}(\mu) = \sum_{p \in S_{q_i}} e^{-\frac{(p-\mu)^2}{2\sigma^2}}, \tag{5.14}$$

where $\sigma \in \mathbb{R}$ is the parameter controlling the smoothing.

Therefore, if we define

$$\nu'_{q_i} = \begin{pmatrix} \nu'_{q_i}(l) \\ \nu'_{q_i}(l+1) \\ \vdots \\ \nu'_{q_i}(r-k+1) \end{pmatrix} \tag{5.15}$$

each sequence x can be mapped into a vector of the form

$$\phi(x) = \begin{pmatrix} \nu'_{q_1} \\ \nu'_{q_2} \\ \vdots \\ \nu'_{q_{4^k}} \end{pmatrix}. \tag{5.16}$$

Our representation for a nucleotide sequence x consists of a vector composed of $[(r-l+1)-k+1] \cdot |\Sigma^k|$ features, each one associated with a particular k-mer $q \in \Sigma^k$ and a specific position in the sequence.

Kernel function (Dual form)

The idea of the *Oligo kernel* [Meinicke et al., 2004, Taher et al., 2004a] is to represent nucleotide sequences by their k-mer occurrences, as vectors composed of functions that preserve only an adjustable degree of information about the location of the k-mers in the sequence. This position uncertainty is achieved by means of a mixture of Gaussian functions with variance σ^2.

Let $\Sigma = \{A, C, G, T\}$ be the DNA alphabet, and x be a nucleotide sequence. Let q be a particular k-mer, $q \in \Sigma^k$, occurring in the sequence x at positions $S_q = \{p_1^q, p_2^q, \ldots\}$. The value of the feature that corresponds to k-mer q at position t contributes to the value of the feature corresponding to k-mer q at position μ according to

$$f_q(t; \mu, \sigma^2) = \begin{cases} e^{-\frac{(t-\mu)^2}{2\sigma^2}}, & \text{if } t \in S_q \\ 0, & \text{otherwise,} \end{cases} \tag{5.17}$$

where $\sigma \in \mathbb{R}$ is the parameter that controls the smoothing.

Let x_1 and x_2 be two nucleotide sequences, q be a k-mer in Σ^k, $S_q^{x_1} = \{p_1^{q,x_1}, p_2^{q,x_1}, \ldots\}$, and

$S_q^{x_2} = \{p_1^{q,x_2}, p_2^{q,x_2}, \ldots\}$ be the set of positions where q occurs in x_1 and x_2, respectively. Then, the product of the two sequence vector representations $\phi(x_1)$ and $\phi(x_2)$ can be written as

$$\begin{aligned}\phi(x_1) \cdot \phi(x_2) &= \int_l^r \sum_{q \in \Sigma^k} \sum_{i \in \mathcal{S}_q^{x_1}} \sum_{j \in \mathcal{S}_q^{x_2}} f_q(t; i, \sigma^2) \cdot f_q(t; j, \sigma^2) dt \\ &= \sum_{q \in \Sigma^k} \sum_{i \in \mathcal{S}_q^{x_1}} \sum_{j \in \mathcal{S}_q^{x_2}} \int_l^r f_q(t; i, \sigma^2) \cdot f_q(t; j, \sigma^2) dt.\end{aligned} \tag{5.18}$$

Let $s = t - i$ and $k = -(i - j)$, then

$$\begin{aligned}\int_l^r f_q(t; i, \sigma^2) \cdot f_q(t; j, \sigma^2) dt &= \int_l^r e^{-\frac{(t-i)^2}{2\sigma^2}} \cdot e^{-\frac{(t-j)^2}{2\sigma^2}} dt \\ &= 2\pi\sigma^2 \cdot \int_{l-i}^{r-i} \frac{e^{-\frac{s^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} \cdot \frac{e^{-\frac{(k-s)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} ds,\end{aligned} \tag{5.19}$$

which is the convolution of two Gaussian functions with equal variances σ^2 and means equal to zero. For two Gaussian functions $N(\mu_1; \sigma_1)$ and $N(\mu_2; \sigma_2)$, their convolution is another Gaussian,

$$N(k; \mu_1, \sigma_1) * N(k; \mu_2, \sigma_2) = \frac{e^{-\frac{[k-(\mu_1+\mu_2)]^2}{2(\sigma_1^2+\sigma_2^2)}}}{\sqrt{2\pi(\sigma_1^2 + \sigma_2^2)}}. \tag{5.20}$$

Replacing 5.20 in 5.19,

$$\begin{aligned}\int_l^r f_q(t; i, \sigma^2) \cdot f_q(t; j, \sigma^2) dt &= 2\pi\sigma^2 \cdot \frac{e^{-\frac{k^2}{4\sigma^2}}}{2\sigma\sqrt{\pi}} \\ &= \sqrt{\pi}\sigma \cdot e^{-\frac{(i-j)^2}{4\sigma^2}}.\end{aligned} \tag{5.21}$$

Therefore, the *kernel function* for x_1 and x_2 can be calculated as follows

$$\phi(x_1) \cdot \phi(x_2) = \sqrt{\pi}\sigma \cdot \sum_{q \in \Sigma^k} \sum_{i \in \mathcal{S}_q^{x_1}} \sum_{j \in \mathcal{S}_q^{x_2}} e^{-\frac{(i-j)^2}{4\sigma^2}}. \tag{5.22}$$

The effect of the smoothing parameter σ can be appreciated in Equation 5.22. The degree of uncertainty regarding the position of the k-mers increases with σ; the case $\sigma = 0$ corresponds to a sequence representation which is strictly position-dependent, while the case $\sigma \to \infty$ produces a sequence representation entirely based on position-independent k-mers.

We have implemented the Oligo kernel in C; the computational cost decreases with the increase of the k-mers length, k, because then the probability of observing the occurrence of the same k-mer in two sequences decreases, and the only positions contributing to the inner product are those which are non-zero in the corresponding feature vectors. The results presented here were obtained with SVM^{Light} [Joachims, 1999], a C implementation of Vapnik's SVM [1995].

5.2.3 Method

Datasets

We employed the HS3D (Homo Sapiens Splice Sites Dataset).

Sequence representation

The nucleotide sequences are represented according to the description given above. We consider sequence windows of 101 nucleotides centred on the first character corresponding to the conserved dinucleotide AG or GT.

Experimental setup

To analyse the splice site sequences by means of the Oligo kernel we first tested the predictive performance of the SVMs for different values of k, σ, and the trade-off between the training error and margin, C, as described in Equation 2.26. We trained classifiers using k-mers of length $k \in \{1,2,3,4,5,6\}$, and $\sigma \in \{0.1, 0.5, 1, 1.5, 2, 2.5, 3\}$. In addition, we selected the optimal C from the set $\{0.1, 0.5, 1, 2, 5, 10\}$.

In order to eliminate any bias caused by a particular partition of training/test sets, we repeated the training and testing process several times using different partitions. More precisely, we used a 3-fold cross-validation scheme; the dataset was divided into three equally-sized, non-overlapping groups, and each cross-validation instance implied training a model using two of these sets, and testing it on the remaining one. In turn, the parameters of the model, i.e., the smoothing parameter σ and the SVM regularisation parameter C, were optimised using a 5-fold cross validation (inside each fold of the outer 3-fold cross-validation). The groups contained an equal number of positive and negative examples. Table 5.17 contains the amount of examples used for training and testing the final models. The sequence window was fixed for all experiments, an equal to $[-50, +50]$.

	Training		Testing	
Signal	Positive Examples	Negative Examples	Positive Examples	Negative Examples
Acceptor sites	1920	1920	960	960
Donor sites	1864	1864	932	932

Table 5.17: Splice site classification using SVMs. Training and testing groups: number of positive and negative examples.

Classification criteria

The performance of the classifier was evaluated in terms of classification rate (CR) and the area under the ROC curve (AUC).

Reference method for comparison of the performance

We compared the performance of SVMs with position-specific weight matrices (WMMs) of order 0, 1 and 2, which have proved to perform particularly well on the problem and are therefore used as reference method [Cai et al., 2000]. We obtained two models, one for the true splice sites (positive class) and one for the pseudo splice sites (negative class), and classified the test examples according to the probability of belonging to either of the classes.

The only parameter that needs to be estimated for these models is the length and position of the window; for a model of order k, the left and right ends of the window were chosen from the set $\{-50, -40, -30, \ldots, +40, +50 - k\}$. The results were obtained using a 3-fold cross-validation scheme, with an internal 5-fold cross-validation scheme for the selection of the optimal window. The training and testing sets coincided with the ones employed for training and testing the SVM.

Discriminative features

As detailed in chapter 2, given a mapping ϕ, a Support Vector Machine with kernel function $K(x_i, x_j) = \phi(x_i)^T \cdot \phi(x_j)$ will classify an instance $x \in \mathbb{R}^n$, in our case, a splice site sequence, according to a function

$$h(x) = w \cdot \phi(x) + b$$

where $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are defined in terms of the solution $\alpha_1^{opt}, \alpha_2^{opt}, \ldots, \alpha_m^{opt}$ of the optimisation problem involving the training examples $x_1, x_2, \ldots, x_m$ and their corresponding labels $y_1, y_2, \ldots, y_m \in \{-1, +1\}$ given by

$$\begin{array}{rc} \max & \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ \text{subject to:} & 0 \leq \alpha_i \leq C, i = 1, 2, \ldots, m, \\ & \sum_{i=1}^m \alpha_i y_i = 0 \end{array}$$

as

$$w = \sum_{i=1}^{m} \alpha_i^{opt} y_i \phi(x_i)^T, \tag{5.23}$$

and

$$b = -\frac{1}{2} \left(\max_{x_i; y_i = -1; \alpha_i > 0} w \cdot x_i + \min_{x_i; y_i = 1; \alpha_i > 0} w \cdot x_i \right). \tag{5.24}$$

From Equation 5.23, it can be noticed that the parameter α_i^{opt} determines the contribution of vector x_i to the training of the SVM model. As the vectors represent the k-mers that are present or absent at specific positions in a nucleotide sequence, α_i^{opt} indicates how significant is the occurrence or absence of a specific k-mer at a certain position for deciding if the sequence corresponds to a splice site or not. Based on this concept, we analyse the *most discriminative features* (i.e., pairs of k-mers and positions) for each SVM classifier.

Visualisation

In order to provide a better insight on the features that are relevant for splice site classification, we define for each position t within the sequence the weight vector $w^*(t)$

$$w^*(t) = \sum_{i=1}^{m} \alpha_i^{opt} y_i [\nu'_{q_1}(t), \nu'_{q_2}(t), \ldots, \nu'_{q_{4^k}}(t)]^T, \tag{5.25}$$

and the matrix

$$W = [w^*(l), w^*(l+1) \ldots, w^*(r-k+1)]. \tag{5.26}$$

W can be visualised as a bitmap image using a colormap to encode the element values or *weights*. The plot of W reveals the k-mers and the positions within the sequence that are relevant for the prediction; features associated with high positive weights correlate well with the positive class, while features associated with high negative weights correlate well with the negative class. We linearly normalised matrix W, so that the weights are comprised between -1 and 1. The *y axis* of the graph corresponds to the k-mers, sorted alphabetically, and the *x axis* represents the position in the sequence, from -50 to $+50-k+1$, where k is the length of the k-mers in the sequence representation, and 0 is the position where the first character of the conserved AG or GT occurs.

k-mer rankings

In the case of higher order k-mers and with the purpose of facilitating the interpretation of the results, we restricted W (as defined in Equation 5.26) to the most discriminative k-mers. We ranked the weight functions corresponding to each k-mer $q_i \in \Sigma^k$ along the complete sequence according to their L_2-norm

$$N_{q_j} = \sqrt{\sum_{t=l}^{r-k+1} \left(\sum_{i=1}^{m} \alpha_i y_i \nu'_{q_j}(t) \right)^2}. \tag{5.27}$$

The ranking was based on the non-normalised values of the matrix. The k-mers with higher L_2-norms contribute the most to the decision function, either through very relevant occurrences at single positions or because these k-mers exhibit unusually high or low values of the weight function overall the sequence.

This procedure ensures a correct and intuitive visualisation of the decision function, facilitating the identification and interpretation of the more relevant features.

When we considered it appropriate, we compared the weights obtained by our SVM model for each k-mer and position in the sequence (each component of the vector presented in Equation 5.25) with simple statistics corresponding to the HS3 database, namely the natural logarithm of the ratio for the frequency of a k-mer at a certain position between real and pseudo splice site sequences (LFR):

$$LFR(q_i, j) = \ln \left(\frac{f^+_{q_i,j}}{f^-_{q_i,j}} \right), \tag{5.28}$$

where $f^+_{q_i,j}$ is the frequency of occurrence of k-mer $q_i \in \{A, C, G, T\}^k$ at position j in *real* splice site sequences, and $f^-_{q_i,j}$ is the frequency of occurrence of k-mer q_i at position j in *pseudo* splice site sequences. As might be expected, employing longer k-mers for this calculation results in problems related to the amount of available data.

Feature selection

The feature subset selection experiments were done using three different algorithms,

- Weight score filter (WSF),
- Recursive Feature Elimination-SVM (RFE-SVM), and
- L_0-norm algorithm.

We use our own MATLAB® implementation of the algorithms. The WSF and RFE-SVM algorithm produce feature rankings, that can be directly compared; the L_0-norm algorithm produces only a final *optimal* feature subset.

We performed the feature subset selection experiments on the sequence representations produced by the Oligo kernel for the optimal smoothing parameters σ obtained in Section 5.2.4. The experiments were done using a 3-fold cross-validation scheme; the selection of the hyperparameter C took place within a 5-fold cross-validation loop. The results presented here correspond to the average of the three cross-validations; the final sets of features analysed are the outcome of the intersection of the three sets corresponding to the cross-validations.

In order to speed up the RFE-SVM procedure, we simultaneously eliminated several features; without this shortcut, the procedure is practically inapplicable. For each sequence representation, we performed exactly five iterations; each iteration involved training a SVM with 100%, 50%, 10%, 1% and 0.1% of the total amount of features, respectively; Table 5.18 lists the number of features involved in each iteration step for the different lengths of k-mers that constitute each sequence representation.

k-mers	100%	50%	10%	1%	0.1%
1-mers	404	202	40	4	0
2-mers	1600	800	160	16	1
3-mers	6336	3168	633	63	6
4-mers	25088	12544	2508	250	25
5-mers	99328	49664	9932	993	99
6-mers	393216	196608	39321	3932	393

Table 5.18: Number of features for each iteration step in the feature selection procedure.

5.2.4 Results

Predictive Performance Experiments

We evaluated SVMs for the prediction of splice site sequences represented by k-mers of lengths 1 to 6 associated with different degrees of positional information, according to a parameter $\sigma \in \mathbb{R}$. The parameter σ and the SVM hyperparameter C that produce the best classification rates (CR) and values for the area under the ROC curve (AUC) are shown in Table 5.19 and Table 5.20; the values presented correspond to the averages of the three cross-validation instances. Figure 5.15 and Figure 5.16 summarise the performance of the SVMs as a function of the length of the k-mers used to represent the sequences, and Figure 5.17 depicts the optimal values of σ as a function of the length of the k-mers.

Acceptor sites

In the case of the acceptor splice sites, the best AUC value is achieved using k-mers of length 5 and a σ value of 2.6 ± 0.2; this value of σ *spreads* the occurrence of a k-mer located at position 0 over the range $[-10, +10]$. The performance observed for k-mers of length 4 is also remarkable; the optimal value of σ is 2.4 ± 0.4. The optimal values obtained for the smoothing parameter σ suggests the existence of motifs that do not occur at a fixed position in the sequence but are relevant for the prediction. The performance of the classifier decreases for sequence representations based on 6-mers, which justifies the use of k-mers up to length 6. The optimal values of σ increase with the value of k; for large values of k this increase is more gradual than for small values of k; monomers are indeed expected to be strictly position-dependent signals, while sequence representations based on longer k-mers might require some flexibility regarding their location.

k	C	σ	CR	AUC
1	0.1 ± 0.0	0.1 ± 0.0	0.877 ± 0.005	0.939 ± 0.001
2	5.9 ± 0.9	0.9 ± 0.0	0.910 ± 0.002	0.960 ± 0.003
3	0.9 ± 0.4	2.2 ± 0.5	0.911 ± 0.009	0.963 ± 0.003
4	0.6 ± 0.0	2.4 ± 0.4	0.920 ± 0.006	0.966 ± 0.002
5	0.63 ± 0.06	2.6 ± 0.2	0.919 ± 0.002	0.969 ± 0.002
6	0.9 ± 0.2	2.73 ± 0.06	0.909 ± 0.003	0.963 ± 0.003

Table 5.19: Acceptor site classification using the Oligo kernel. Optimal parameters according to the AUC value.

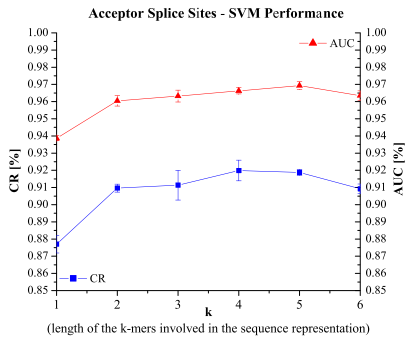


Figure 5.15: Acceptor site classification using the Oligo kernel. Best performing models.

Donor sites

Regarding the prediction of donor splice sites, the best AUC value is achieved using sequence representations based on 3-mers and 5-mers, with σ values of 0.45 ± 0.05 and 1.0 ± 0.1, respectively; these values *spread* signals over three and seven positions, respectively; the performance of the classifier is also good for 4-mers, and, in general, for all the values of k employed in this study, but the classification rates and AUC values decrease for sequence representations based on very short ($k = 1, 2$) and very long ($k = 6$) k-mers. Compared to acceptor splice sites, the optimal values of σ increase only gradually with k, suggesting that an accurate prediction of donor splice sites strictly requires position-dependent information, and indicating the existence of a quite long highly conserved region. The worst performance is achieved by position-dependent monomers, which insinuates that the classification relies on *conditional* position-specific occurrences of nucleotides.

k	C	σ	CR	AUC
1	5 ± 4	0.15 ± 0.05	0.918 ± 0.005	0.969 ± 0.003
2	0.7 ± 0.3	0.3 ± 0.1	0.932 ± 0.006	0.972 ± 0.002
3	0.7 ± 0.1	0.45 ± 0.05	0.941 ± 0.002	0.976 ± 0.001
4	1.5 ± 1	0.9 ± 0.3	0.940 ± 0.009	0.975 ± 0.003
5	3.7 ± 0.9	1.0 ± 0.1	0.944 ± 0.006	0.976 ± 0.004
6	5 ± 1	1.3 ± 0.4	0.935 ± 0.008	0.974 ± 0.004

Table 5.20: Donor site classification using the Oligo kernel. Optimal parameters according to the AUC value.

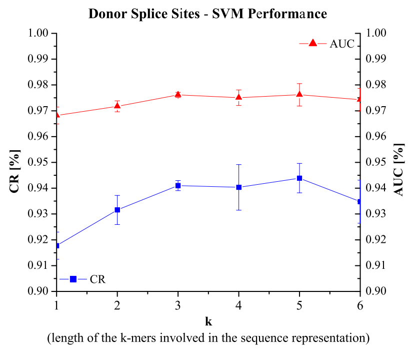
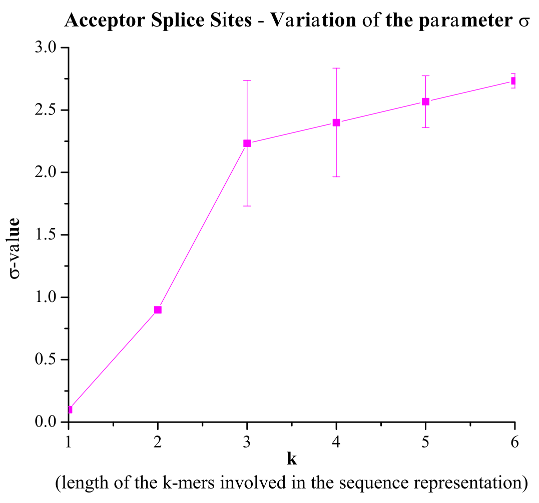


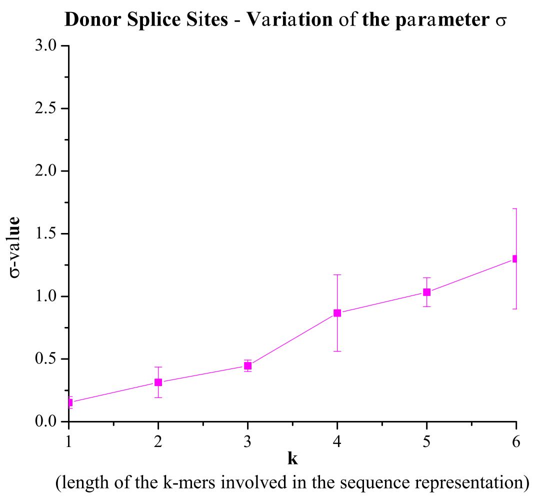
Figure 5.16: Donor site classification using the Oligo kernel. Best performing models.

Reference method for comparison

In the case of acceptor sites, the optimal average over all-runs position-specific weight matrices has order 2 and produced an AUC value of 0.954 ± 0.001 on a window located at $[-17, +1]$. For donor sites, the optimal position-specific weight matrix has order 2 and an AUC value of 0.967 ± 0.008 when evaluated on the window $[-6, +10]$. The classification rate is in both cases close to 0.9. The



(a)



(b)

Figure 5.17: Values of the smoothing parameter σ corresponding to the best performing models for acceptor sites (a) and donor sites (b).

performance of the optimal position-specific weight matrix models is comparable both in term of the AUC values and the classification rates achieved to that of our SVM models, supporting the validity of our approach. The parameters corresponding to the best models are summarised in Table 5.21.

Signal	order	Window	CR	AUC
Acceptor sites	2	[-17 $\pm$ 1, 1 $\pm$ 1]	0.894 $\pm$ 0.005	0.954 $\pm$ 0.001
Donor sites	2	[-6 $\pm$ 0, 10 $\pm$ 0]	0.92 $\pm$ 0.01	0.967$\pm$ 0.008

Table 5.21: Splice site prediction using position-specific weight matrix models (reference).

Discriminative features

As an indication of the discriminative power of each feature, we used the components of the weight vector produced by the best performing SVM models (Equation 5.25); each component or *feature* corresponds to a pair (q_i, j), where q_i is a k-mer in $\{A, C, G, T\}^k$ and j is a position in the sequence. For features corresponding to short k-mers, we compare our findings with the *LFR* in Equation 5.28; in addition, we examine differences and similarities with regard to the consensus regions informed by the literature, *(C/T)N(C/T)**AG**(A/G)* for acceptor sites and $AG\boldsymbol{GT}(A/G)AGT$ for donor sites (the characteristic conserved dinucleotides are signalled in bold), and the pyrimidine-rich region which is known to precede acceptor sites, as described in chapter 1. The positions reported here are with respect to the occurrence in the sequence of the first character of the acceptor site AG or donor site GT, which corresponds to position 0.

Acceptor sites

The sequence representations that performed best for the classification of acceptor sites were based on 4-mers and 5-mers. The ten most discriminative features for the sequence representation based on tetramers correspond to occurrences of $TTTT$ between positions -16 and -7, with a maximum weight at position -11; in this sense, the presence of the tetramer $TTTT$ in the upstream region seems to be highly significant for appropriately distinguishing between real and pseudo acceptor sites. The same is true for sequence representations relying on pentamers; in this case, the ten most discriminative features are the occurrence of $TTTTT$ from position -18 to -9, with a weight maximum at position -13. The next most relevant tetramer feature for the classification of acceptor sites is the occurrence of $CAGG$ at position -1, highlighting the importance of the occurrence of the nucleotides C and G at positions -1 and $+2$, respectively; other pentamer features with high weights include $TGCAG$ at position -3 and $CAGGT$ at position 0; these features are in agreement with the known consensus region. In terms of their absolute values, features with positive weights are more discriminative than features with negative weights.

Sequence representations relying on trimers also perform well. The non-normalised distribution of the trimer weights, calculated according to Equation 5.25 is presented in Figure 5.18. The feature with maximal weight is the occurrence of TTT at position -11; indeed, the occurrence of TTT is a very strong signal upstream of the AG, starting at position -22. Other features with high positive scores are CAG at position -1, and AGG at position 0, in the consensus region, and a series of features associated with the pyrimidine-rich region upstream of the acceptor site, represented by the occurrences of TTC with weight peak at positions -4 and -3, TCT at position -7 and CCC at position -6. The ten features with lowest weights are the occurrences of TTT between positions $+33$ to $+40$, with a weight minimum at position $+37$, and the occurrences of GAG at position -1; the last feature indicates that in acceptor site sequences, G usually does not precede the conserved dinucleotide AG, and evidences the relevance of the conserved region for distinguishing real acceptor sites from pseudo acceptor sites. We clustered the top 100 features according to their positions; from these, 77 occur in the upstream region, between positions -27 and -3, and the remaining 23 occur between positions -2 and $+2$; the features in the interval $[-27, -3]$ correspond mainly to occurrences of the 3-mer TTT. Although the occurrences of *T-rich* k-mers is probably related to the known pyrimidine-rich region that is present right upstream of the conserved AG, the predominance of *T-rich* k-mers around position -11 appears to be a particularity of real acceptor sites. The remaining features reflect the effect of the smoothing on the consensus region; between -2 and $+2$, CAG occurs with maximum weight at position -1, AGG at position 0, and GGT at position $+1$, with a consensus $CAGGT$.

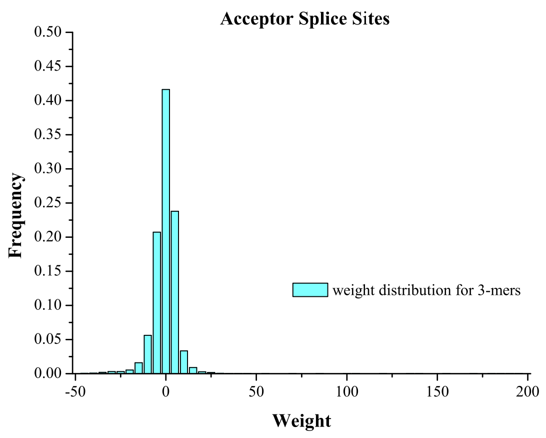


Figure 5.18: Acceptor site classification using the Oligo kernel. Weight distribution for trimers.

The importance that TTT, $TTTT$, and $TTTTT$ have for the correct classification of acceptor sites using SVMs suggests the existence of a *T-rich* region upstream of the AG in real acceptor site sequences, in addition to the pyrimidine-rich region which is known to directly precede the AG; this characteristic can also be noticed for shorter k-mers (see Figure 5.19). Other features that are clearly important for the classification of acceptor sites using SVMs are the aforementioned pyrimidine-rich region and the consensus region.

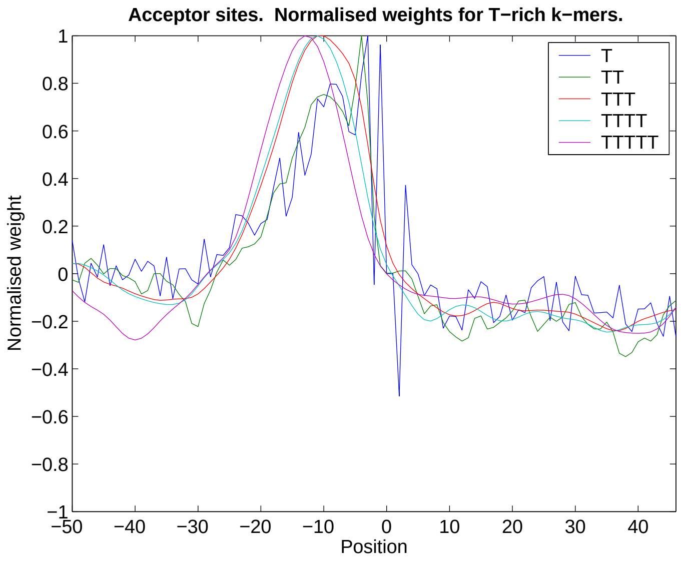


Figure 5.19: Acceptor site classification using the Oligo kernel. Detail of the weight functions for T-rich k-mers.

Donor sites

True donor sites are characterised by the occurrence of AGT at position +3. Other trimer features with very high weights include GAG at position +2, AGG at position −2, GGT at −1, GTA at positions 0 and +4, AAG at position +2, TAA and TGA at position +1 and GTG at position +4; all these features are connected to the consensus region, and suggest a consensus $AGGT(A/G)AGT$ starting at position −2, which matches the reported consensus region. The features with the lowest weights are GTT and GTC at position 0; they are followed by TGT and AGT at position −1, and TGG, TGT, TTT, TCC at position +1, and CGT and GGG at positions −1 and −2, respectively; the highly negative weights might be explained by the differences that these k-mers show with respect to the known consensus. The distribution of the weights is shown in Figure 5.20. From the top 100 features, 73 are located downstream of the GT signal. In particular, k-mers like CCC and GCC seem to be relevant immediately downstream of the consensus region, while GGG predominates downstream of position +10; these results are confirmed by the difference in the frequencies between real and pseudo donor site sequences observed to these k-mers. The signals in the upstream region have low weights and are indeed difficult to interpret.

Concerning tetramers, the ten features with highest weights are located between positions −2 and +2; the most important feature is the occurrence of $GGTA$ at position −1, followed by other features that also match the consensus region. However, the occurrence of $GGTA$ at positions −2 and 0 is also among this group; these features result from the degree of smoothing, but are clearly incompatible with the consensus region, which probably explains the increase in the standard deviation for the predictive performance of the SVMs using sequence representations based on tetramers.

Finally, the most important pentamers for the classification of donor sites using SVMs in terms of their scores are located between positions −3 and +1, and also confirm the consensus sequence $(A/C)AG\boldsymbol{GT}(A/G)AGT$. The good prediction rates observed for donor site representations including pentamers are clearly based on this consensus region, and the fact that the prediction rates decrease for longer k-mers is likely to indicate relations between different sections of the complete 9-nucleotide-long consensus region, which only shorter k-mers are able to capture appropriately.

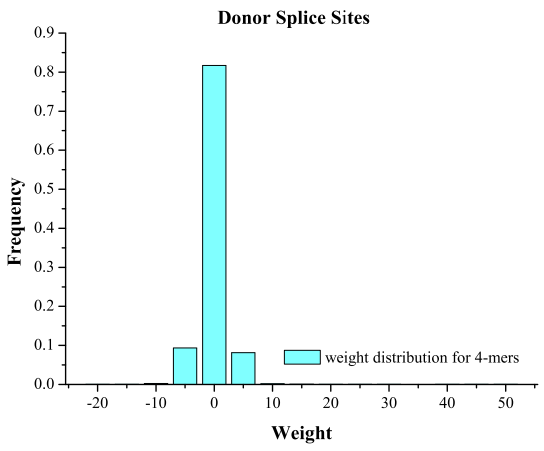


Figure 5.20: Donor site classification using the Oligo kernel. Weight distribution for trimers.

In addition, as indicated by the absolute values of their weights, features with negative weights are considerably less discriminative than features with positive weights.

Visualisation

The decision functions of the best classifiers were visualised using an overview bitmap image of matrix W in Equation 5.26. The rows in the image correspond to the different k-mers, while the columns describe their positions (l to $r - k + 1$, where $-l = r = 50$) within the sequence; the colour of a pixel indicates the value of the corresponding feature. The images produced in this way facilitate the obtention of conclusions, such as those presented in the preceding section.

Figure 5.21 shows the 1024x97 pixel image obtained from the acceptor site model produced by the optimal average parameters σ and C, using pentamers, the k-mers that proved to perform best. The *y axis* contains all possible pentamers, from $AAAAA$ at the bottom to $TTTTT$ at the top; the *x axis* indicates the position in the sequence, with respect to the occurrence of the conserved AG.

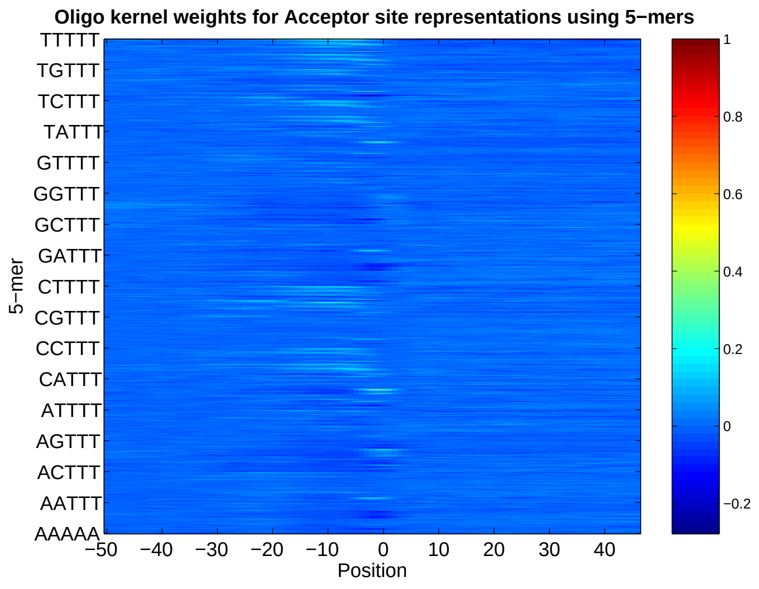


Figure 5.21: Acceptor site classification using the Oligo kernel. Classification using pentamer features.

Figure 5.22 shows the 64x99 pixel image obtained from the donor site model produced by the optimal average parameters σ and C, using trimers. The *y axis* contains all possible trimers, from AAA at the bottom to TTT at the top; the *x axis* indicates the position in the sequence, with respect to the occurrence of the conserved GT. The most discriminative features are k-mers located in close proximity to the conserved GT. The strongest features are coloured from dark red to orange, and correspond to AGT at position +3, GAG at position +2, AGG at −2, GGT at −1, GTA at 0 and +4, AAG at +2, TAA at +1, and TGA at +1; these signals correspond exactly to the known consensus region, *(A/C)AG***GT***(A/G)AGT*.

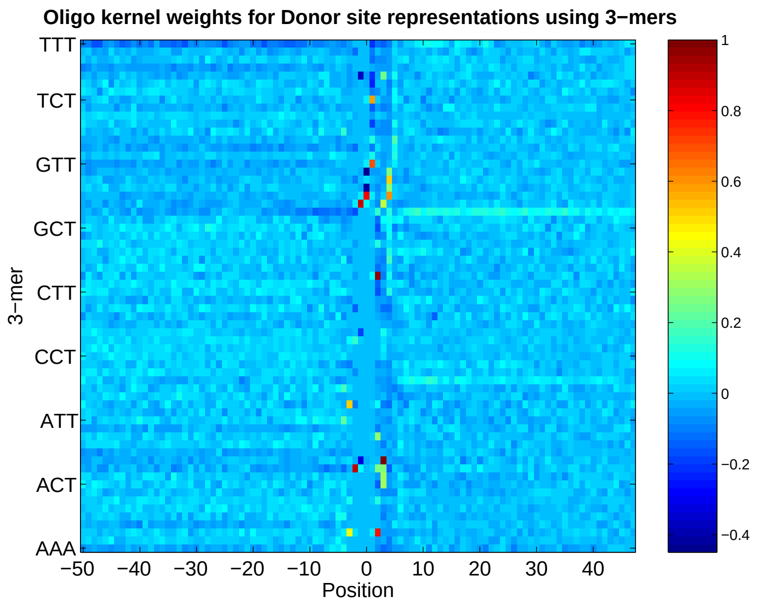


Figure 5.22: Donor site classification using the Oligo kernel. Classification using trimer features.

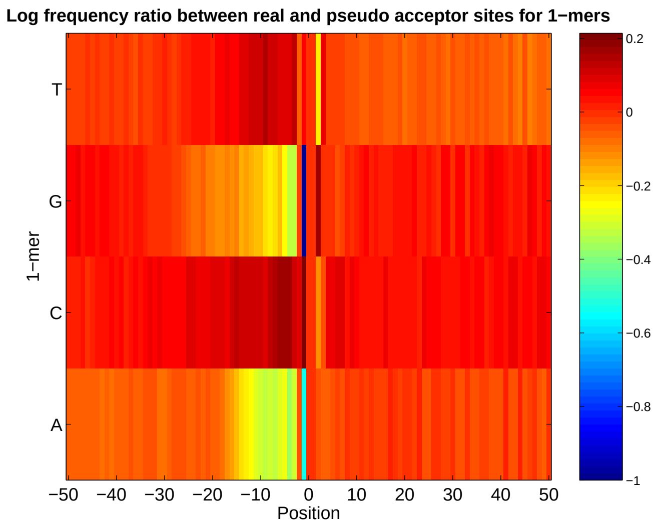
k-mer Rankings

We identified the ten most discriminative k-mers according to the L_2-norm of their weights over the complete nucleotide sequence, as described in Equation 5.27, using the models produced by the optimal parameters. The ten most significant k-mers are depicted as bar plots in Figures 5.25 and 5.28; the height of the bars is linearly scaled so that the maximum is one.

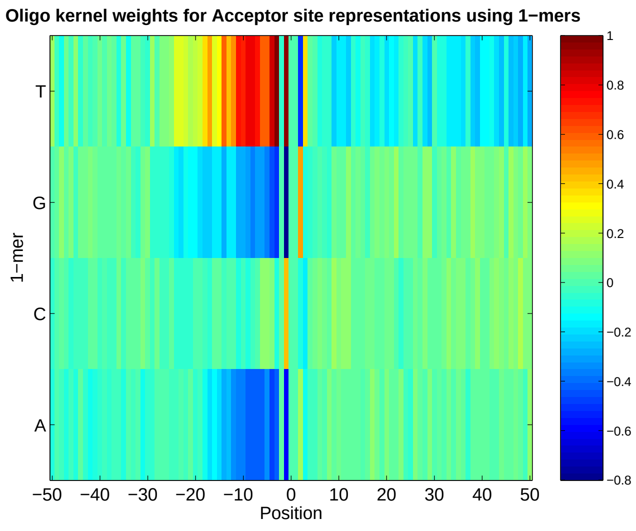
Acceptor sites

The most discriminative monomer in acceptor site prediction is T, followed by G, A, and C, in decreasing order. When observing real and pseudo acceptor site sequences, C and T present particularly high *log frequency ratios* upstream of the conserved AG; nevertheless, they show slightly different behaviours; C frequency ratios increase gradually, with a maximum at position −4, while T's increase only starting approximately at position −25, reaching its maximum around position −11, and decreasing abruptly downstream of the AG; such peak is compatible with a signal that occurs not in a fixed position, but in a range of positions, while the behaviour of C suggests a preference for this nucleotide regarding the general composition of the sequence. This hypothesis explains the differences observed for the comparison between the *LFRs* and the weight matrix produced by the SVM in Figure 5.23, and supports our monomer ranking.

In the case of dimers, TT is followed by AG, GA, CA, GG, GC, AA, CG, TA, and TG. TT is preferred at position −2, and also within a range of positions upstream of the conserved dinucleotide AG, centred at −10; on the contrary, AG is particularly less frequent in these regions.



(a)



(b)

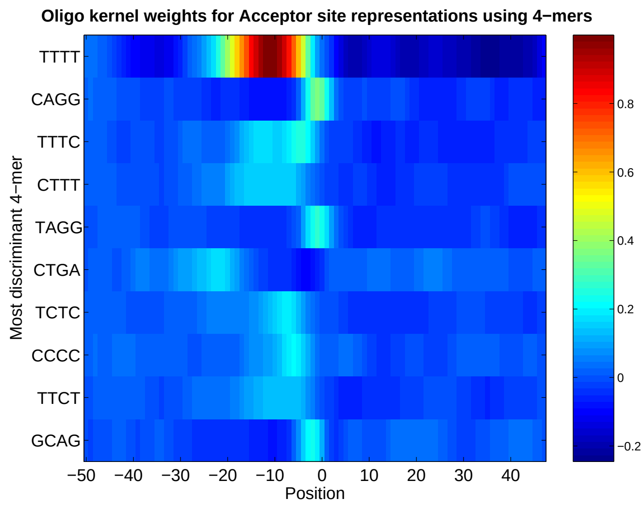
Figure 5.23: Acceptor site classification using monomer features. Comparison between the log frequency ratios between real and pseudo acceptor site sequences (a) and the normalised SVM weights (b). The colormaps reflect the different scales.

From the next five dimers, three are positively associated with the consensus region of acceptor sites, *(C/T)N(C/T)**AG**(A/G)*, and the remaining two, *GA* and *AA*, are negatively associated at position -1, confirming that this position is likely to correspond to a pyrimidine. Finally, *CG* presents slightly more positive weights than the average in the downstream region, and *TA* and *TG* present minimum weights in positions upstream of the consensus region. All these patterns correspond very well with the *log frequency ratios* calculated for the data, validating the use of the weights produced by our SVM models as an indication of biological meaning.

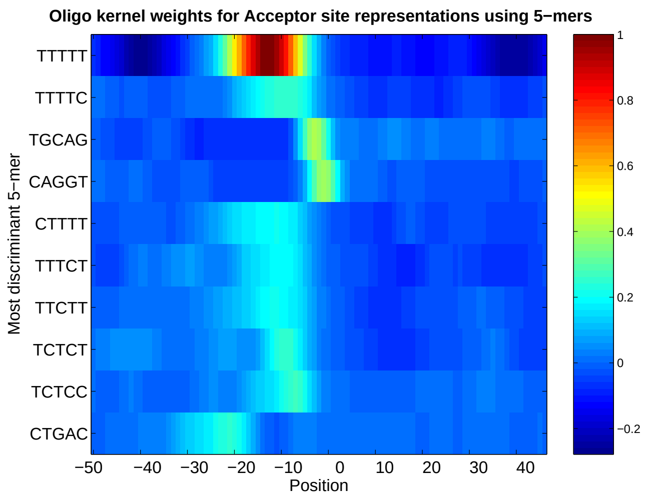
The top 10 trimers of the corresponding ranking can be divided into three categories; *TTT* shows a maximum weight at position -11; there are eight trimers associated with the consensus region *(C/T)N(C/T)**AG**(A/G)* between positions -3 and $+2$, and among these, *GAG*, *AAG*, and *GGA* are negatively associated, with minimum weights at positions -1, -1 and -3, respectively; finally, *TCT* represents the pyrimidine-rich region that precedes the conserved *AG*, with a weight maximum around position -8.

Among the most discriminative tetramers and pentamers, there are four well-defined groups. The tetramer with the highest discriminative power is *TTTT*; *CAGG* and *TAGG* at position -1, and *GCAG* at position -2 contribute to the prediction mainly because they correspond to the consensus region; a series of features related to the pyrimidine-rich region are also associated with high weights, in the upstream region; *CTGA* is likely to be related to the branch site. *TTTTT* appears as the most discriminative pentamer, with maximum weights around position -13, and is followed by pentamers that can be associated to the pyrimidine-rich region, and two pentamers, *TGCAG* and *CAGGT* which corresponds to the known consensus region. The weights associated for these k-mers at each position of the sequence are depicted in Figure 5.24.

Finally, we note that nine of the ten top hexamers are directly related to the consensus region, highlighting its importance; the last two differ from the consensus at position $+3$, with a *G* and a *T*. The occurrence of *TTTTTT* shows a minimum around position -25, clearly different from k-mers with average behaviour; the analysis of the *LFR* observed for this hexamer in real and pseudo donor sites, suggests that they occur more often in the latter at a series of upstream positions, which suggests



(a)



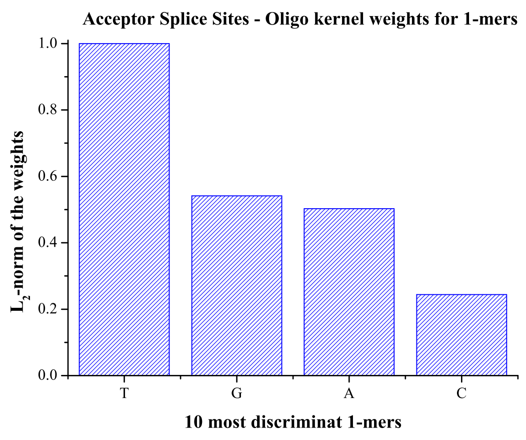
(b)

Figure 5.24: The graph show, sorted from top to bottom, the most discriminative 4-mers (a) and 5-mers (b) for the classification of acceptor sites.

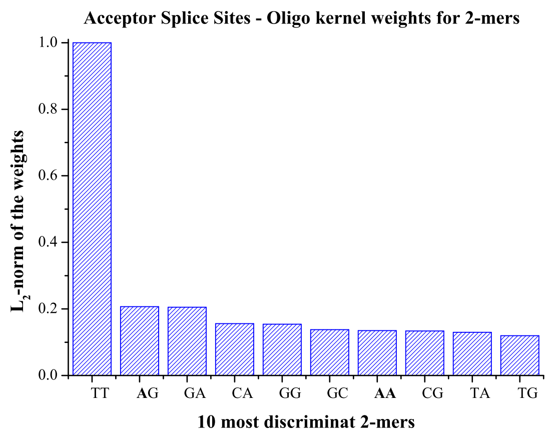
some degree of positional-flexibility; this would explain the increase in the optimal σ value.

Among the ten hexamers with highest discriminative power, $TTTTTT$ is followed by four k-mers corresponding to the pyrimidine-rich region that occurs right upstream of the AG and five k-mers with maximum weights at position -4, directly related to the consensus region, that indicate the preference for pyrimidines at positions -4, -3 and -1.

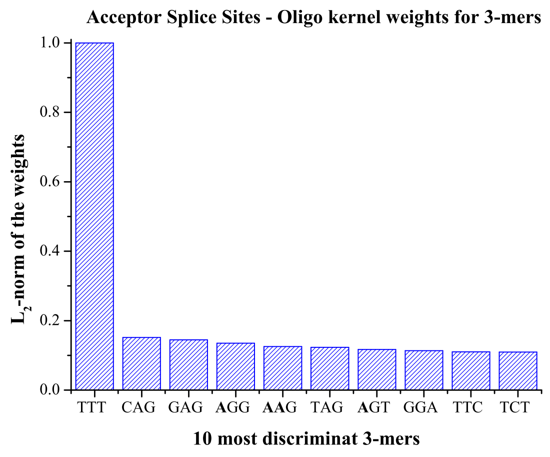
Based on the previous observations, we can conclude that the good prediction rates with which the SVM is able to separate real acceptor sites from pseudo acceptor sites are due to four kinds of features. For all values of k, we found that k-mers in $\{T\}^k$ are the most relevant k-mers, and we can argue that this is due to a short region with a high density of T nucleotides upstream of the conserved AG whose exact location varies. In addition, the SVM clearly recognises the occurrence of a pyrimidine-rich region upstream of the conserved AG as a feature that characterises real acceptor site sequences. Finally, the consensus region and the branch site are also recognised as important features, but their extensions seem to be limited.



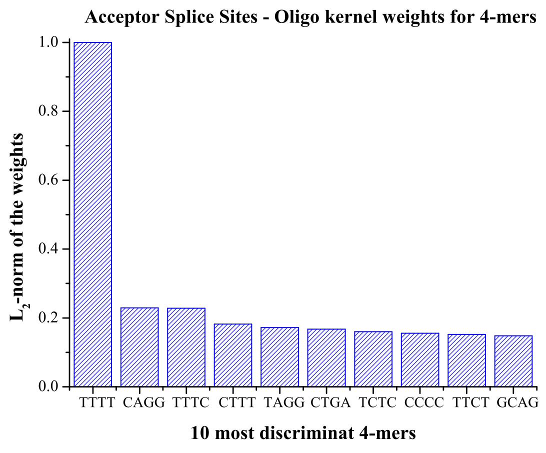
(a)



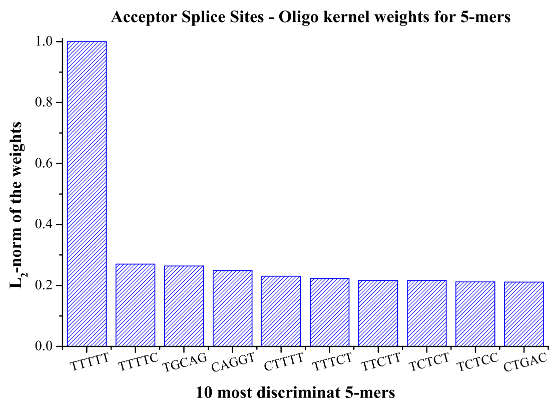
(b)



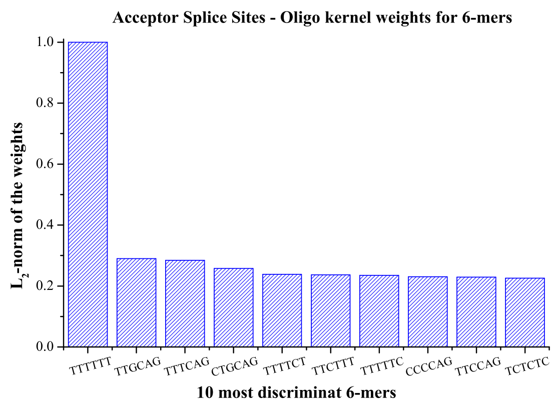
(c)



(d)



(e)



(f)

Figure 5.25: The bar plots present the ten most discriminat k-mers for the classification of acceptor sites, according to their SVM weights: (a) 1-mers, (b) 2-mers, (c) 3-mers, (d) 4-mers, (e) 5-mers, (f) 6-mers.

Donor sites

For the prediction of donor sites, T is the most discriminative monomer, followed by G, A, and C, in decreasing order. T is also the most important monomer when comparing the monomer frequency ratios between real and pseudo donor sites. As it can be observed in Figure 5.26 (a), the GC content is slightly increased both upstream and downstream of real donor sites, a feature appropriately captured by the SVMs (Figure 5.26 (b)). Regarding T, its occurrence in the upstream region is associated with pseudo donor sites rather than with real donor sites.

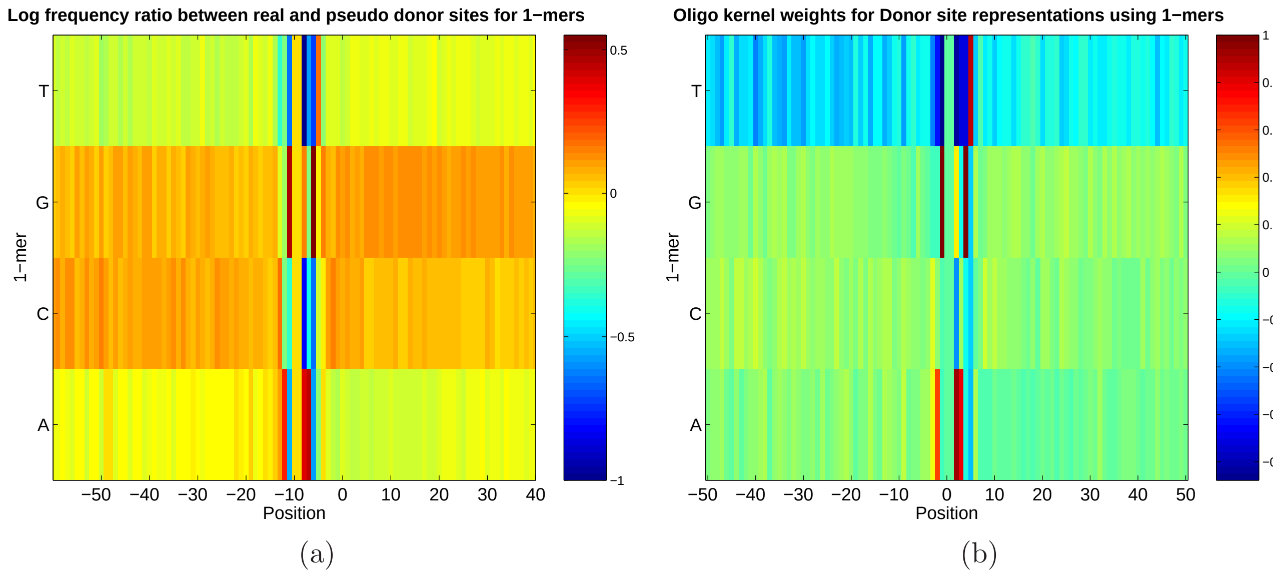


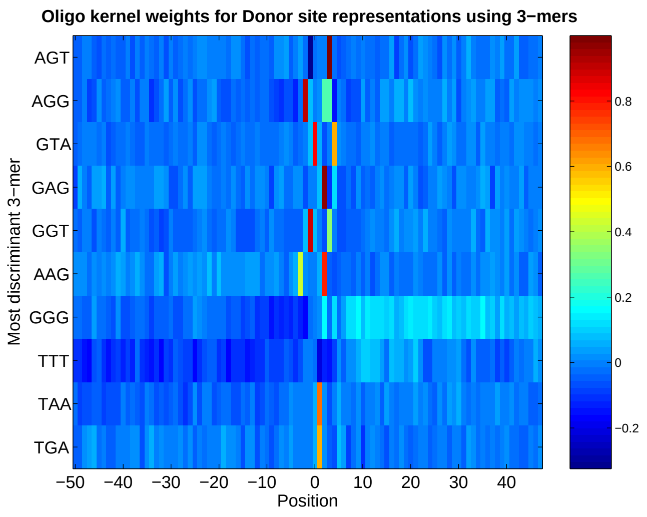
Figure 5.26: Donor site classification using monomer features. Comparison between the log frequency ratios between real and pseudo donor site sequences (a) and the normalised SVM weights (b). The colormaps reflect the different scales.

Except for the dimer TT, all relevant dimers, AG, GG, TA, GT, GA, AA, TG, TC, CA, are exclusively related to the consensus region; in particular, TG and TC present negative weights at positions +2 and 0, respectively, and TT exhibits a negative weight at position +2 and weights more negative than the average upstream of the conserved dinucleotide GT.

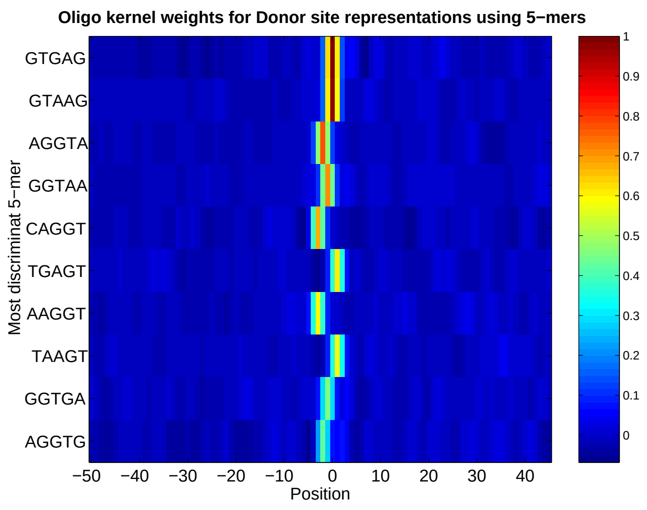
Regarding tetramers, the most relevant ones are clearly related to the consensus sequence; the exception is $TTTT$, which has weights slightly more negative than the average tetramers in the upstream region, and slightly more positive in the downstream region, with a maximum around position +15; the negative weights in the upstream region correspond with a more abundance of $TTTT$ observed in pseudo donor sites, when compared to real donor site sequences.

The sequence representations that achieve the best classification rates, composed by trimers and pentamers, apparently owe their prediction power to k-mers reconstructing the consensus sequence *(A/C)AG**GT**(A/G)AGT*. Additionally, in the case of trimers, TTT shows weights slightly smaller than the average in the upstream region, while GGG shows weights slightly larger than the average in the downstream region, and this can also be noticed when observing the frequency of occurrence of these trimers in the sequences. Figure 5.27 presents the weights for the ten most discriminative trimers and pentamers.

Finally, we note that nine of the ten top hexamers are directly related to the consensus region, highlighting its importance; the last two differ from the consensus at position +3, with a G and a T. The occurrence of $TTTTTT$ shows a minimum around position −25, clearly different from k-mers with average behaviour; the analysis of the frequencies observed for this 6-mer in real and pseudo



(a)

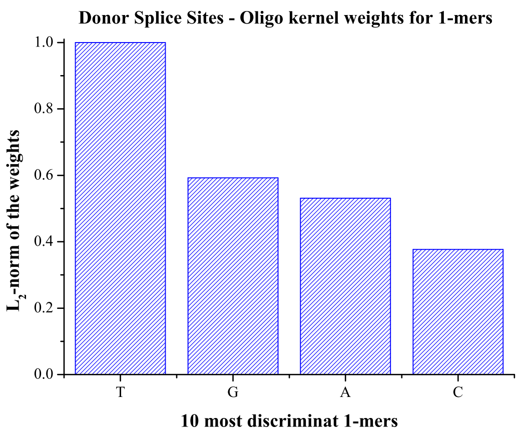


(b)

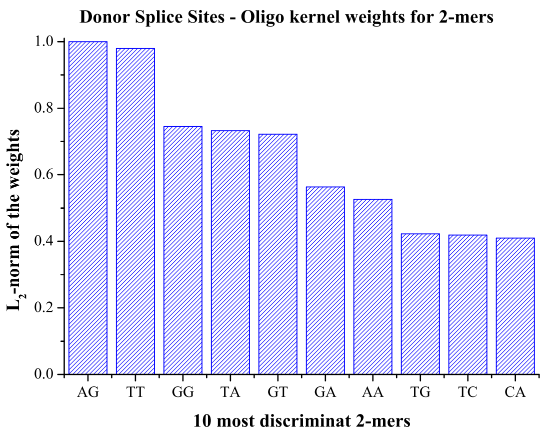
Figure 5.27: The graph show, sorted from top to bottom, the most discriminative 3-mers (a) and 5-mers (b) for the classification of donor sites.

donor sites, suggests that they occur more often in the latter at a series of upstream positions, which suggests some degree of positional-flexibility; this would explain the increase in the optimal σ value.

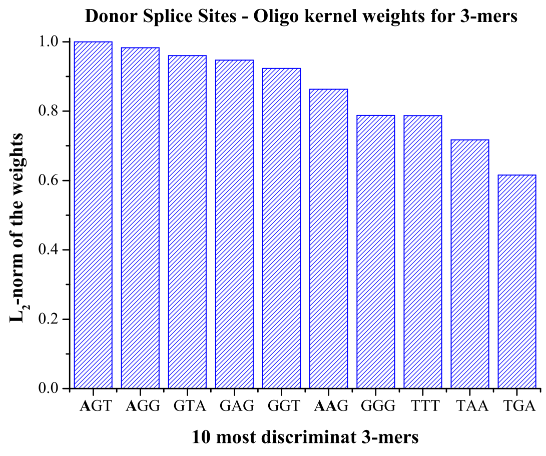
These results suggest that the good prediction rates achieved for the prediction of donor sites are mainly due to the nucleotides in the known consensus region; however, the results also suggest interactions between nucleotides in the aforementioned region, which are responsible for the variation of the prediction rates with the length of the k-mers chosen for the sequence representation. There is also evidence for a role by k-mers with a high content of the nucleotide T, which seem to be less frequent in the upstream region of real donor site sequences; this is confirmed by analysing the logarithm of the frequency ratios (LFR) between real and pseudo donor site sequences, and can be partly explained by the fact that coding sequences tend to have higher GC content than non-coding sequences; nevertheless, as it can be observed for different k-mer lengths, it is possible that the occurrences of T in the upstream region have actually a particular role, related to the nucleotide composition requirements for the sequence.



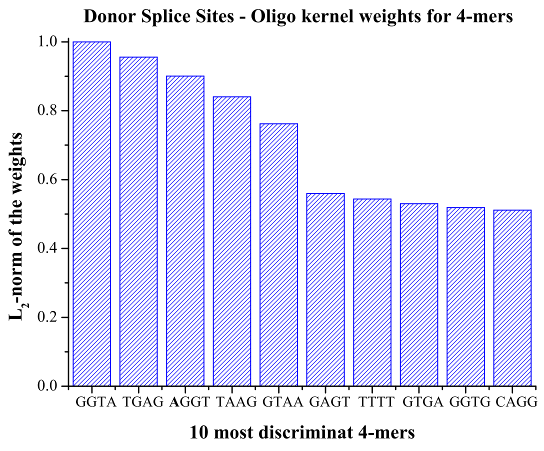
(a)



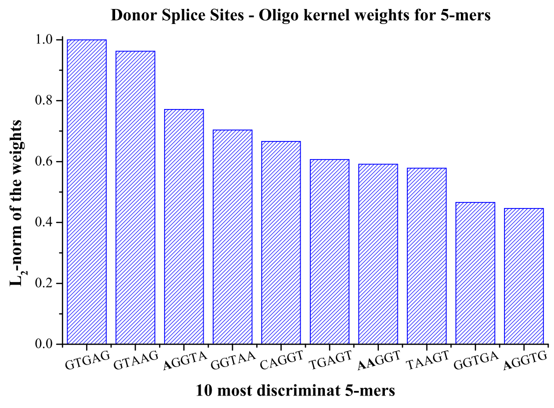
(b)



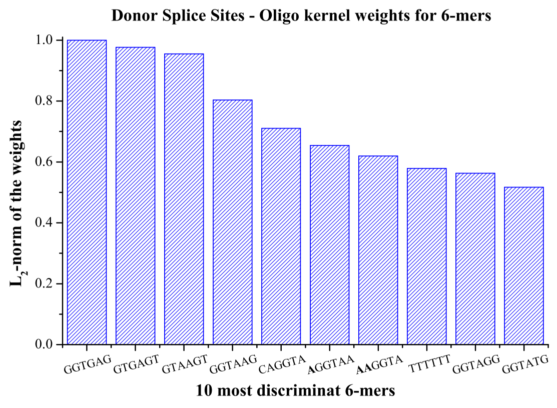
(c)



(d)



(e)



(f)

Figure 5.28: The bar plots present the ten most discriminat k-mers for the classification of donor sites, according to their SVM weights: (a) 1-mers, (b) 2-mers, (c) 3-mers, (d) 4-mers, (e) 5-mers, (f) 6-mers.

5.2.5 Feature subset selection

In order to corroborate and complement the feature analysis that we presented in section 5.2.4, we compared three different feature subset selection methods, the RFE-SVM algorithm, the Weight Score Filter (WSF), and the L_0-norm algorithm.

The comparison involved training and testing the SVM with different sets of features, as described above, and then evaluating the performance of the SVM. The rankings produced by the RFE-SVM algorithm and the WSF were compared directly; on the contrary, the optimal subset of features produced by the L_0-norm algorithm was considered separately; the later algorithm does not produce a ranking and is not applicable to all k-mer representations, as it requires the positive and negative classes to be linearly separable.

The changes in the predictive performance for different subsets of features obtained by the WSF and the RFE-SVM algorithm are presented in Figures 5.29 and 5.30, respectively. These results are averages over three cross-validations.

In general, the SVM is not able to perform well when only a very low percentage ($\leq 0.1\%$) of the total number of features is used; under those circumstances, all instances are predicted either as positive or as negative. The results show that when only a very low percentage ($\leq 0.1\%$) of the total number of features is used, the RFE-SVM subsets perform better than the WSF subsets; in most other cases, the prediction power of the WSF features is superior. Feature selection using the RFE-SVM algorithm does not result in an improvement of the performance, possibly due to the rough iteration feature groups chosen; however, considering the size of the datasets, other schemes would result infeasible in terms of the running times. Employing features selected by the weight score filter results in a slight improvement when about 50% of the total number of features is used for the classification. The analysis of the most significant features is complex, as the three sets obtained from all cross-validations differ extensively, probably indicating a high degree of correlation among the features; the intersection of these three sets is always larger for the features obtained with the weight score filter.

The performance of the classifier using the subset provided by the L_0-norm algorithm was generally worse than the performance of the classifier using a similar amount of features and any of the other two feature selection methods mentioned; the features obtained were related to the ones reported by the RFE-SVM algorithm, which is expected considering the similar nature of both methods, but the size of the final sets was highly dependent on the convergence criterion. We omit the presentation of the data because of the poor performance of the corresponding classifiers.

Acceptor sites

The prediction rates of the SVMs using the top 10% features of the WSF monomer and dimer rankings are superior to the prediction rates obtained using the top 10% features of the RFE-SVM ranking. The sets of features resulting from the intersection of the three sets produced by the three cross-validation instances contain 22 and 25 elements, respectively, in the case of the WSF, and none in the case of the RFE-SVM algorithm. These sets include monomers and dimers upstream and downstream from the conserved AG, but do not include any relevant feature in the consensus region, which probably indicates the existance of many correlated features, which do not need to be captured all to ensure a good prediction power.

For trimers, the top 10% produced by the WSF is distributed along the whole window sequence, while the features produced by RFE-SVM are with few exceptions upstream of the AG. The RFE-SVM top features clearly include a pyrimidine-rich region between positions -25 to -4, and suggest a special abundance of trimers mainly composed of T between positions -12 and -9. Trimers associated with the consensus region (at positions -1 to $+2$) are also reported as relevant. The features produced by the WSF are more varied and difficult to interpret.

Regarding tetramers, while the intersection of the top 1% produced by the WSF is empty, the intersection of the top 1% produced by RFE-SVM contains eight elements, all signals related to the consensus region except for the occurrence of $TTTC$ at position -13. However, for the same amount of features, the top WSF features exhibit more predictive power than the top features in the RFE-SVM ranking, suggesting that the WSF captures additional relevant features. We observe similar results for longer k-mers; the intersection of the top 1% of the features in the tetramer RFE-SVM ranking contains 19 features, composed by pyrimidine-rich k-mers upstream of the AG and k-mers related to the conserved region; the top 10% of the features in both RFE-SVM and WSF ranking includes features located along the complete window. For long k-mers, the feature subsets produced by the algorithms are complex, and identitfying the most significant features is not trivial.

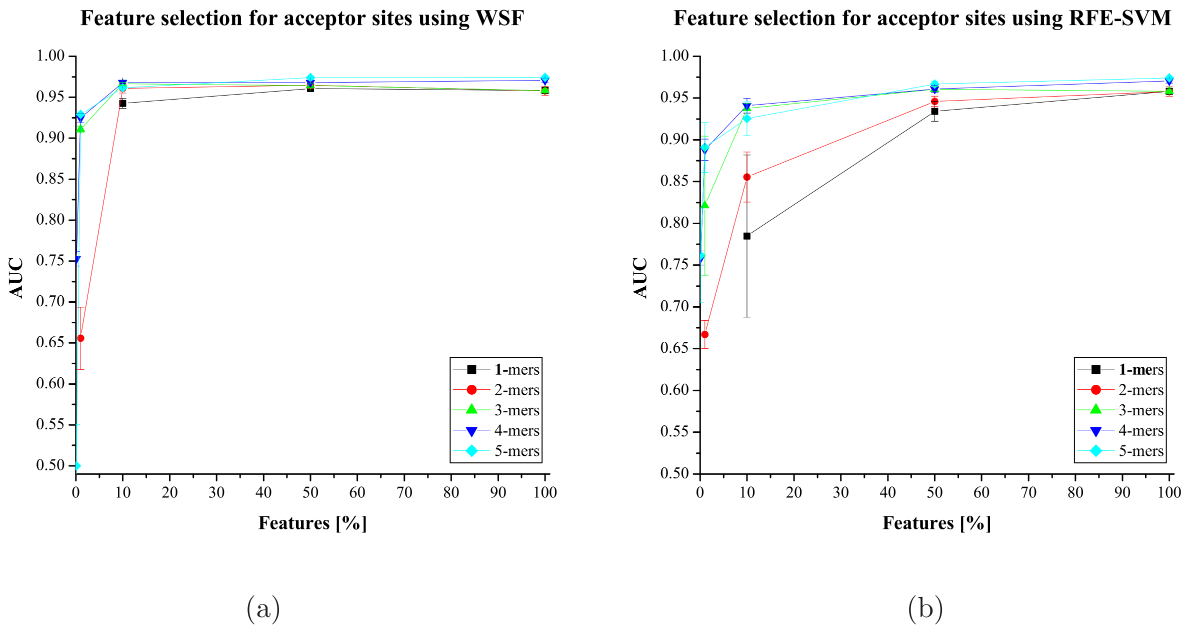


Figure 5.29: Acceptor site classification using SVMs. AUC values for SVM classification with different feature subset, using the WSF (a) and the RFE-SVM algorithm (b).

This study confirms the findings made with the Oligo kernel. We can conclude that the nucleotides flanking the acceptor site are certainly important for the predictive performance, but not the only features responsible for it. Longer k-mers show an overall more discriminative power, and they are required not at a fixed position, but in a relatively broad range of positions. The existence of a pyrimidine-rich region upstream of the AG conserved dinucleotide is a very strong feature. In addition, the results indicate that in part of this region, between positions -13 and -9, T is preferred over C. The feature subsets obtained by the different methods confirm that most relevant features for acceptor site prediction are situated upstream the AG, but the results obtained with the WSF suggest the correlation of these features with features occurring downstream of the acceptor site, which might obey to geometrical (complementarity) requirements of the splicing mechanism.

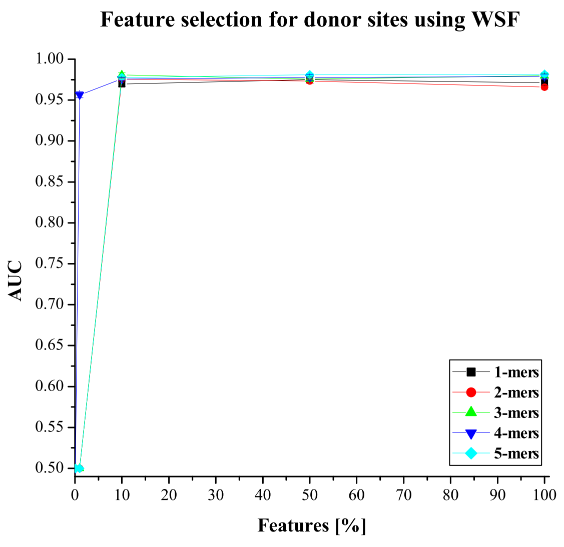
Donor sites

As in the case for acceptor sites, the SVM performs better using the top 10% features of the WSF monomer and dimer rankings than the top 10% features of the RFE-SVM ranking. The set of features resulting from the intersection of the three sets produced by the WSF in the three cross-validation instances contain 20 and 16 elements, respectively, and none in the case of the RFE-SVM algorithm. These sets include monomers and dimers upstream and downstream from the conserved GT, but not between positions -4 and $+6$; the same is true for trimers, suggesting many correlated features.

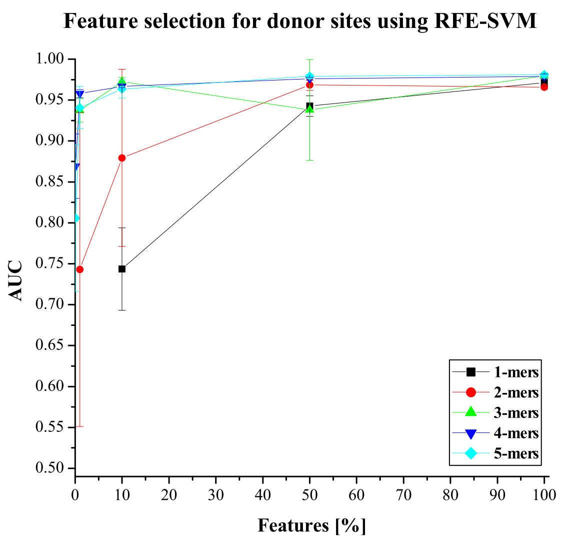
For longer k-mers, the top 10% features in the WSF ranking, which show more predictive power than the top features in the RFE-SVM ranking, the interpretation is difficult; they do not include features within the known conserved region. On the contrary, the top features in the RFE-SVM rankings mainly include features in the consensus region, and k-mers with a high content of G and C downstream of the conserved GT. The worse performance of the RFE-SVM rankings might be due to redundancy in the feature sets.

The predictive performance for donor sites decreases fast when a very low percentage of the total features is used. While the RFE-SVM ranking is relatively robust to the reduction of features, the WSF ranking is not.

The strongest features seem to be the ones around the consensus dinucleotide GT, there is also evidence of the relevance for certain k-mers usptream and downstream; the difficulty to identify them suggests actually a change on the average nucleotide composition, likely to correspond to the difference between coding and non-coding regions.



(a)



(b)

Figure 5.30: Donor site classification using SVMs. AUC values for SVM classification with different feature subset, using the WSF (a) and the RFE-SVM algorithm (b).

5.2.6 Discussion and conclusions

We have described how to use an SVM kernel to systematically study and characterise acceptor and donor splice site sequences. Many features identified by our method as the most discriminative ones are in strong correspondence both with previous knowledge about splice site sequences and simple statistics distinguishing real from pseudo splice site sequences in the dataset. These findings support the hypothesis that all features identified with means of the method described here have in fact biological significance, and therefore the analysis of the features' influence on the decision function of an SVM can enrich our understanding of biological sequences. Our SVM approach is a generalisation of previous studies, and as such, able to provide a more informative overview of the splice site prediction problem.

We conclude that the prediction of acceptor sites depends basically on the following sets of features: position-dependent monomers in the consensus region and branch site, pyrimidine-rich region immediately upstream of the AG, and occurrence of T-rich k-mers around position -10. Although the consensus region is clearly strictly positional-dependent, the representation of the other two kinds of features can be improved by incorporating certain position uncertainty, as presented here. The prediction of donor sites seems to rely almost exclusively on position-dependent nucleotide occurrences, which would in turn be correlated; in addition, our results show that there exist other features, like the absence of k-mers with a high content of the nucleotide T in the upstream region of real donor sites. These results should be considered in the design of prediction tools.

On the other hand, standard feature selection methods do not add any additional valuable information regarding the discriminative features, and in this case, seem to be less informative than the direct study of the decision function of the classifier.

5.3 Splice Site Prediction using Gaussian-smoothed Weight Matrix Models

5.3.1 Introduction

Markov chain and weight matrix models have been widely used for studying genomic sequences and shown to be well suited for modelling and predicting splice sites [Cai et al., 2000]. Using as training set the nucleotide sequence in the neighbourhood of known splice sites (and on certain occassions also of pseudo splice sites), the goal is to estimate a model that is able to accurately predict splice sites on unseen data. The models consist basically of a matrix, whose elements represent the conditional probabilities of observing certain bases in the sequence, assuming a particular series of preceding bases; the number of preceding bases considered is referred to as order. Higher-order models perform normally better than lower-order models, but unfortunately, these higher-order models have a number of limitations associated with high state-space complexity and reduced coverage, resulting sometimes on an even worse overall prediction accuracy.

Hereby, we propose a heuristic to improve the prediction performance of higher-order position-specific weight matrices (WMMs). The key idea behind our method is the smoothing of the probabilities of the weight matrix models using the Gaussian function; by assuming that the probabilities of contiguous positions in the sequence are not completely independent, we greatly reduce the number of zero-probabilities without arbitrarily having to add pseudo-counts to all positions. Even though the method was developed in the context of splice site prediction, it might be certainly useful for different applications.

5.3.2 Method

Datasets

The experiments presented here were performed on the HS3D sequences.

5.3.3 Standard weight matrix models

Let $(X_1, X_2, X_3, \ldots, X_l)$ be a sequence of nucleotides of length l, with $X_i \in \Sigma = \{A, C, G, T\}$. A standard $0th$-order position-specific weight matrix model is characterised by a matrix $M = (m)_{ij}$, with $m_{ij} = P(X_i = k_j)$ representing the probability of finding nucleotide $k_1 = A$, $k_2 = C$, $k_3 = G$, or $k_4 = T$ at position i in the sequence; matrix $M = (m)_{ij}$ contains l rows (positions in the sequence) and 4 columns (one for each nucleotide). The probability of a sequence of nucleotides x according to model M is

$$P(x|M) = \prod_{i=1}^{l} m_{ij_i},$$

where j_i is the matrix column (1 for A, 2 for C, 3 for G and 4 for T) corresponding to the nucleotide found at position i in the nucleotide sequence x.

The classification of a given sequence x of length l is determined by the log-odds ratio between the model for real splice site sequences, M^+, and the model for pseudo splice site sequences, M^-.

$$S = \log\left(\frac{P(x|M^+)}{P(x|M^-)}\right).$$

Position-specific weight matrix models of order $k-1$ consider sequences of k-mers from the set $\Sigma^k = \{A, C, G, T\}^k$ labelled according to an index set I instead of sequences of simple nucleotides (monomers). The matrix that characterises such system contains $l-k+1$ rows (positions in the sequence) and 4^k columns (one for each k-mer). In this case, P is not a probability measure. Let A be the event given by the *sequences in which k-mer with index i occurs at position i*, and B be the event given by the *sequences in which k-mer with index j occur at position j*, with $i + 2(k-1) < j$ or $j + 2(k-1) < i$; these two events are clearly disjoint. However, as the calculation of $P(A \cup B)$ involves the multiplication of f_{ii} and f_{jj}, respectively, by all its prefixes and suffixes of lengths $k-1$, $k-2$, …, 1, $P(A \cup B) \neq f_{ii} + f_{jj}$ and P does not satisfy the additivity axiom. Nevertheless, P is related to a probability measure by means of a normalisation constant.

The probability of a sequence x of length l given the normalised model M is

$$P^*(x|M) = \prod_{i=1}^{l-k+1} \frac{m_{ij_i}}{c_{ij_i}}$$

where $c_{ij_i} \geq 0$ are the normalisation constants.

Let M^+ and M^- be the positive and the negative models, respectively, the log-odds ratio determines the classification of a sequence s according to

$$\log\left(\frac{P^*(s|M^+)}{P^*(s|M^-)}\right) = \log \frac{\prod_{i=1}^{l-k+1} \frac{m^+_{ij_i}}{c^+_{ij_i}}}{\prod_{i=1}^{l-k+1} \frac{m^-_{ij_i}}{c^-_{ij_i}}} = \sum_{i=1}^{l-k+1} \log \frac{m^+_{ij_i}}{m^-_{ij_i}} - \sum_{i=1}^{l-k+1} \log \frac{c^+_{ij_i}}{c^-_{ij_i}} = \log\left(\frac{P(s|M^+)}{P(s|M^-)}\right) + Z,$$

for an appropriate normalisation constant Z.

5.3.4 Gaussian-Smoothed weight matrix models

We propose to smooth the matrix M by means of the Gaussian function

$$g(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{5.29}$$

where μ is the mean and σ is the standard deviation of the Gaussian distribution. The value for each element in the new matrix $P = (p)_{ij}$ is then defined by

$$p_{ij} = \sum_{k=1}^{l} g(k; i, \sigma) \cdot m_{kj}. \tag{5.30}$$

In order to do that efficiently, we first compute a smoothing matrix S based on the Gaussian function. The (i, j)-*th* element of this matrix is given by

$$s_{ij} = \frac{e^{-\frac{(i-j)^2}{2\sigma^2}}}{c_i}, \tag{5.31}$$

where $i = 1, 2, \ldots, l$ and $j = 1, 2, \ldots, l$ represent the positions in the sequence, σ is the parameter controlling the degree of smoothing, and c_i is a normalisation constant. The columns of matrix S need to be subsequently normalised, so that their sum is one. For each $j = 1, 2, \dots, l$

$$\sum_{i=1}^{l} s_{ij} = 1.$$

Let $M = (m)_{ij}$ be the matrix of the original system, with m_{ij} representing the probability of finding the k-mer $\in \{A, C, G, T\}^k$ indexed as j starting at position i in the sequence. The values m_{ij} can be estimated from the sequence training set, by measuring for each k-mer and each position in the sequence the frequency of occurrence of k-mer j at position i, f_{ij}.

Finally, the new matrix for the Gaussian-smoothed weight matrix model, P, is given by the product between the transposed of S and M

$$P = (p)_{ij} = S^T \cdot M. \tag{5.32}$$

P defines our Gaussian-smoothed position-specific weight matrix models.

For $\sigma \to 0$, S is the identity matrix, and $P = M$; in that case, the model corresponds to a standard position-specific weight matrix.

5.3.5 Experimental setup

We trained and tested position-specific weight matrices of order 0, 1, 2, 3 and 4, with degree of smoothing $\sigma \in \{0.2, 0.4, \dots, 2\}$. In addition, all the experiments were performed using three windows over the sequence: the window proposed by Salzberg [1997] ($[-14, +2]$ for acceptor sites, and $[-3, +14]$ for donor sites), $[-25, +25]$, and $[-50, +50]$. In all cases, 0 corresponds to the position where the first nucleotide of the consensus dinucleotide AG or GT occurs.

The results presented here correspond to the average over 50 runs on random partitions of the sequence dataset. We separated the data into three sets, two of whom were used for the parameter selection and the remaining one for testing; the sizes of these two sets are listed in Table 5.22; all the available sequences in the datasets were employed, resulting in unbalanced sets that correspond to a realistic splice site prediction setup. The selection of the model parameter σ was achieved using a 5-fold cross-validation scheme: the data above referred to as training data was split in turn ten times into a training composed of 80% of the data (about 0.53% of the complete dataset) and a testing set composed of the remaining 20% (about 0.13% of the complete dataset). The best performing σ-value obtained through the cross-validation experiments was subsequently used to train a new model and test it on the unseen 1/3 of the data.

	Training		Testing	
Signal	Positive Examples	Negative Examples	Positive Examples	Negative Examples
Acceptor sites	1920	219,574	960	109,787
Donor sites	1864	181,288	932	90,644

Table 5.22: Splice site classification using WMMs. Size of the training and testing sets for which the results are reported.

5.3.6 Reference method for comparison

The performance of the models obtained with Gaussian-smoothed weight matrices was primarily compared to the performance of models obtained with standard weight matrices (i.e., without any smoothing). Standard weight matrices models are characterised by a matrix $P = (p)_{ij} = S^T \cdot M$, where $S = (s)_{ij}$ is the identity function,

$$s_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise.} \end{cases} \tag{5.33}$$

Furthermore, we compared our best models with standard weight matrices whose probabilities were smoothed by the addition of pseudo-counts, by means of the following operator

$$S = (1 - lc)I + cO, \tag{5.34}$$

where l is the length of the sequence, I is an $l \times l$ identity matrix, O is an $l \times l$ matrix of $1s$, and $c \leq \frac{1}{l}$ is the parameter that determines the influence of the pseudo-counts. The final model is given by matrix $M = S^T \cdot M$. The values of c were selected from the set $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$.

The experiments were performed according to the training and testing scheme employed for the evaluation of our model.

5.3.7 Wilcoxon Sum Rank test

We used the Wilcoxon Sum Rank test [Wilcoxon, 1945] to detect significant differences between the classification results obtained by our method and the reference methods.

Let A and B be two samples of sizes n_A and n_B, respectively, which have been independently drawn from two populations. The Wilcoxon rank-sum test is a non-parametric significance test, based upon ranking the $n_A + n_B$ observations of the combined sample. The smallest observation gets rank 1, while the largest observation gets rank $n_A + n_B$. The sum of the ranks for the first sample W is the *Wilcoxon rank sum statistic*. If the two populations have the same distribution, then W has mean

$$\mu_W = \frac{n_A(n_A + n_B + 1)}{2} \tag{5.35}$$

and standard deviation

$$\sigma_W = \sqrt{\frac{n_A n_B (n_A + n_B + 1)}{12}}. \tag{5.36}$$

The test rejects the hypothesis that the two populations have identical distributions when W is *far* from its mean. The distribution of W becomes approximately normal as n_A and n_B increase. This approximation can be used to calculate a $p-$value. The $p-$value of a one-sided Wilcoxon sum rank test, which evaluates the significance of the *difference* between two samples, is $P(W \geq x)$; the lower the $p-$value, the greater the probability that the observed differences between pairs of observations in samples A and B is significant.

5.3.8 Classification criteria

The bias in the number of examples of each class makes the classification rate an irrelevant criterion for the performance, so we only report the area under the Receiver Operator Characteristic curve, or AUC value.

5.4 Results

5.4.1 Influence of the window length

As a preliminary study, we examined the influence of the window length on the performance of the method. Nevertheless, the window is not treated as a parameter for the model. The results are summarised in Figure 5.31.

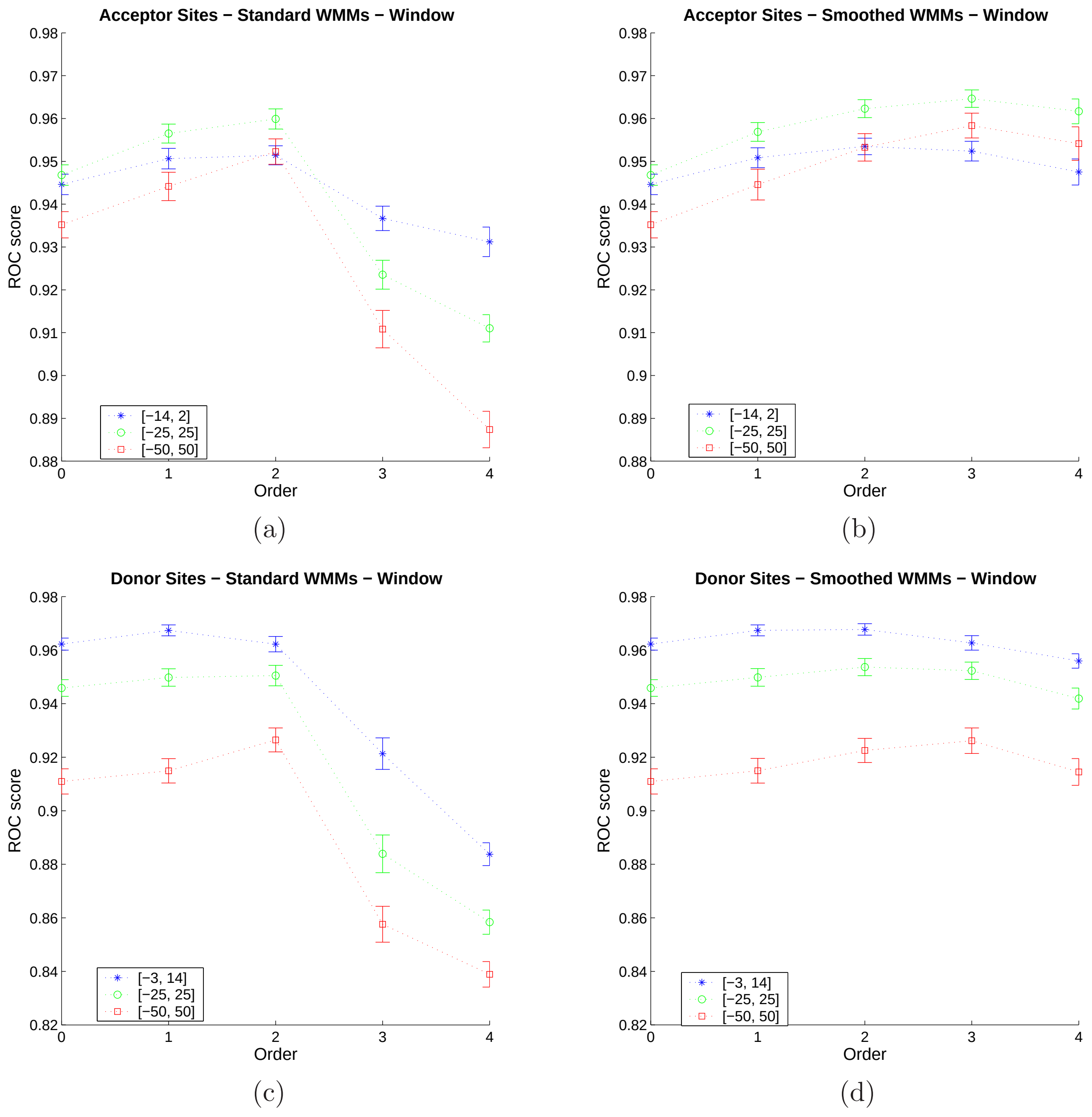


Figure 5.31: Splice site classification using WMMs. Performance for different windows: (a) Acceptor Sites - Standard WMMs, (b) Acceptor Sites - Gaussian-smoothed WMMs, (c) Donor Sites - Standard WMMs, (d) Donor Sites - Gaussian-smoothed WMMs

In the case of acceptor splice site Gaussian-smoothed WMMs, the best performances are always observed with window $[-25, +25]$. Besides, the performance of the method decreases more steeply for window $[-14, +2]$ than for any of the other two longer windows as the order increases. Standard WMMs present a similar behaviour up to order 2; then, the best performances are achieved with the shortest window.

The best window for donor sites is $[-3, +14]$. The performance improves relatively faster for longer windows as the order increases, but for standard WMMs falls dramatically for orders higher than 2. Very long windows ($[-50, +50]$) produce the worst results.

High-order WMMs are expected to require more coverage in order to work well, and therefore, longer sequence windows. But longer windows imply an increase of the number of states with probabilities close to zero, which in turn results in a worse performance. For standard WMMs, the influence of such *zero probabilities* is clearly stronger for orders higher than 2. The use of Gaussian-smoothed position-specific weight matrices greatly reduces the amount of zero probabilities, and increases the efficacy of longer windows.

In the case of donor sites, the results indicate that the most relevant signals are located in the known consensus region, and that any extension of the window where the model is calculated has a negative effect on the performance of the classifier.

As a conclusion, the results presented here were calculated on the window $[-25, +25]$ for acceptor sites, and on the window $[-3, 14]$ for donor sites.

5.4.2 Prediction performance

Acceptor sites

The results presented in Tables 5.23, 5.24 and 5.25, and showed in Figure 5.32, reveal that the advantages of the smoothing become more evident as the order of the model increases. The best overall model corresponds to Gaussian-smoothed WMMs of order 3, with a σ-value of 0.82 ± 0.07. The performance of this model was compared to the performance of the best performing standard (non-smoothed) WMM, i.e., of order 2, and to the performance of the best WMM smoothed with pseudo-counts. The significance of the results were evaluated in terms of the Wilcoxon sum rank test.

order	AUC value
0	0.947±0.002
1	0.956±0.002
2	0.960±0.002
3	0.924±0.003
4	0.911±0.003

Table 5.23: Performance of standard WMMs for acceptor site prediction.

order	σ	AUC value
0	0.2±0.00	0.947±0.002
1	0.20±0.01	0.957±0.002
2	0.39±0.04	0.963±0.002
3	0.82±0.07	0.965±0.002
4	1.96±0.04	0.962±0.003

Table 5.24: Performance of Gaussian-smoothed WMMs for acceptor site prediction.

Based on the Wilcoxon sum-rank significance test, the AUC values obtained with Gaussian-smoothed WMMs are significantly higher than the ones obtained with standard WMMs with *p-value* equal to $3.7 \cdot 10^{-14}$. Regarding the performance of the best model produced with position-specific weight matrices smoothed with pseudo-counts, the results obtained with our Gaussian-smoothed WWM of order 3 are significantly higher with *p-value* $= 0.0456$.

order	c	AUC value
0	1.0E-06±1E-07	0.947±0.003
1	1.5E-06±9E-07	0.958±0.003
2	9E-06±8E-06	0.964±0.002
3	0.0003±0.0002	0.963±0.002
4	0.001±0.000	0.96±0.002

Table 5.25: Performance of WMMs with pseudo-counts for acceptor site prediction.

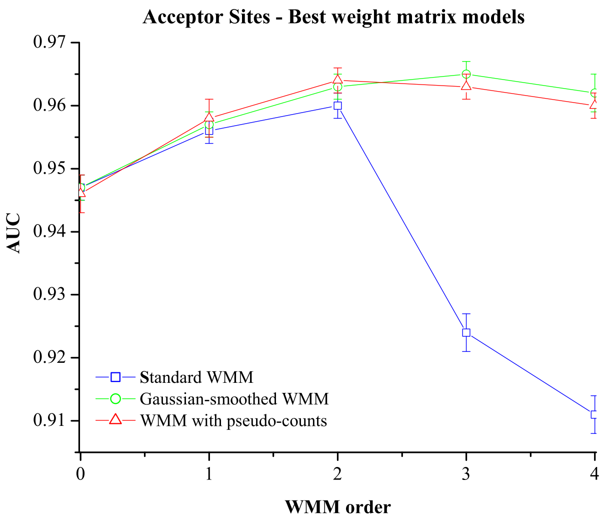


Figure 5.32: Performance comparison of the best weight matrix models for acceptor sites.

Donor sites

Tables 5.26 and 5.27 and Figure 5.33 present the AUC value comparison between Gaussian-smoothed WMMs and standard WMMs for donor sites. The best Gaussian-smoothed weight matrix model corresponds to order 2 and $\sigma = 0.38 \pm 0.03$. The best standard WMM has order 1. We compared both models using the Wilcoxon sum rank test.

order	AUC value
0	0.962±0.002
1	0.967±0.002
2	0.962±0.003
3	0.921±0.006
4	0.884±0.004

Table 5.26: Performance of standard WMMs for donor site prediction.

order	σ	AUC value
0	0.201±0.006	0.962±0.002
1	0.21±0.02	0.967±0.002
2	0.38±0.03	0.968±0.002
3	0.65±0.06	0.963±0.003
4	1.68±0.06	0.956±0.003

Table 5.27: Performance of Gaussian-smoothed WMMs for donor site prediction.

order	c	AUC value
0	1.1E-06±4E-07	0.962±0.004
1	4E-05±4E-05	0.967±0.002
2	0.0004±0.0002	0.968±0.002
3	0.001±0	0.966±0.002
4	0.001±0	0.96±0.002

Table 5.28: Performance of WMMs with pseudo-counts for donor site prediction.

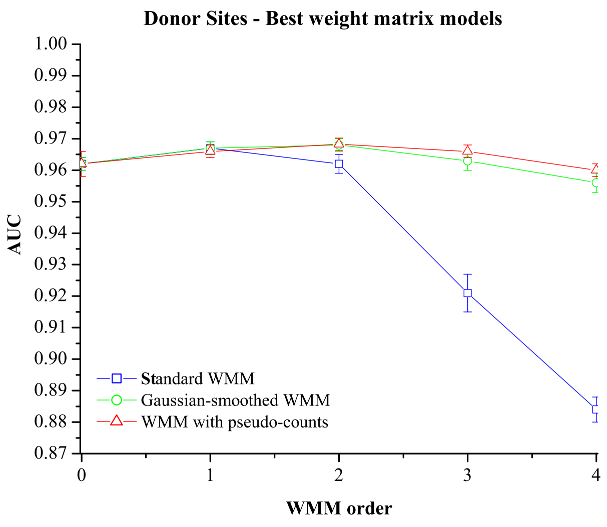


Figure 5.33: Performance comparison of the best weight matrix models for donor sites.

The results obtained with Gaussian-smoothed WMMs are significantly higher than the ones obtained with non-smoothed WMMs only with $p \geq 0.2040$; In other words, the mean of the sample represented by the best Gaussian-smoothed model and the mean of the sample represented by the best standard weight matrix model do not differ with a confidence level higher than 0.7960, which is usually not considered sufficient. Likewise, with respect to the comparison of the $2nd$-order Gaussian-smoothed model and the best performing WMM smoothed with pseudo-counts, the *p-value* obtained (0.2 for the null hypothesis "the distributions have different means") indicates that both models perform equally.

5.4.3 Combinations of WMMs

Finally, we tested the performance of simple combinations of weight matrix models.

Acceptor sites

With the same window we used for the previous experiments, $[-25, +25]$, we tested combinations of the best Gaussian-smoothed weight matrices, i.e., $3rd$-order weight matrices, with $0th$-, $1st$-, $2nd$- and $4th$-order models. We also trained these models with $\sigma \in \{0.2, 0.4, \ldots, 2, 2.2\}$. The performances achieved with the different combinations are presented in Table 5.29. The combinations of standard Markov chains did not perform better than the best simple models, with AUC values inferior or equal to 0.965.

Model	orders	σ		AUC value
1	0 & 3	0.2±0.0	0.75±0.04	0.964±0.001
2	1 & 3	0.2±0.0	0.76±0.05	0.965±0.001
3	2 & 3	0.36±0.05	0.8±0.1	0.965±0.002
4	4 & 3	2.16±0.04	0.683±0.04	0.965±0.002

Table 5.29: Evaluation of the best performing weight matrix model combinations for acceptor site prediction.

The best performing combinations of weight matrices (models 2, 3 and 4) were compared to the best performing model obtained so far for classifying acceptor sites, i.e., Gaussian-smoothed $3rd$-order weight matrices. The Wilcoxon sum rank test did not find the difference to be significant with confidence levels higher than 75%, which means that the combination of models does not improve the prediction performance.

Donor sites

Also employing the window $[-3, +14]$, we combined the best performing Gaussian-smoothed weight matrix, i.e., $2nd$-order models, with $0th$, $1st$, $2nd$, and $3rd$-order Gaussian-smoothed WMMs, with values of $\sigma \in \{0.2, 0.4, \ldots, 2, 2.2\}$. As expected, the optimal σ-values resulted to be similar to the ones obtained of the individual models. The best performing combination of standard weight matrices produces an AUC value of 0.968.

Nevertheless, as the Wilcoxon sum rank test proves, the performance of this model is only significantly higher than the one obtained for standard $1st$-order weight matrices for confidence levels lower than 91%.

Model	orders	σ		AUC value
1	0 & 2	0.2±0.0	0.4±0.0	0.968±0.002
2	1 & 2	0.21±0.02	0.38±0.03	0.968±0.002
3	3 & 2	0.68±0.03	0.28±0.03	0.966±0.002
4	4 & 2	1.73±0.05	0.29±0.05	0.964±0.002

Table 5.30: Evaluation of the best performing weight matrix model combinations for donor site prediction.

5.5 Discussion

5.5.1 Acceptor sites

The the maximum performance for standard weight matrices is observed for weight matrices of order 2; models with higher than 2 do not describe acceptor sites appropriately. The optimal performance for Gaussian-smoothed weight matrices is obtained for models of order 3.

We can conclude that Gaussian-smoothed $3rd$-order WMMs are more suitable models than lower order WMMs. The smoothing allows estimating the distribution for bases in a specific context, given a small sample of bases from that distribution; its effect is the decrease in the amount of zero-probabilities that is normally the consequence of higher order representations. As demonstrated with the Wilcoxon sum rank test, the increase in the performance produced by the smoothing is significant. The behaviour observed may also suggest the occurrence of signals that are important for the prediction outside the known consensus region.

5.5.2 Donor sites

The best model for standard weight matrices corresponds to order 1; the best model for Gaussian-smoothed weight matrices has order 2 and a relatively small value of σ. The means of the AUC values for both models have been shown to be statistically equal, implying that the smoothing does not result in a significant improvement of the performance, and suggesting that donor site signals are highly positional dependent. Indeed, for higher orders, WMMs with pseudo counts perform best, with relatively large values for the parameter c.

As expected, both for acceptor and donor sites the optimal values of σ increase with the order of the model, implying that in order to produce a good prediction rate, the position of longer words must be delocalised to longer intervals (see Figure 5.34).

The combination of weight matrix models does not produce better performances than the individual models.

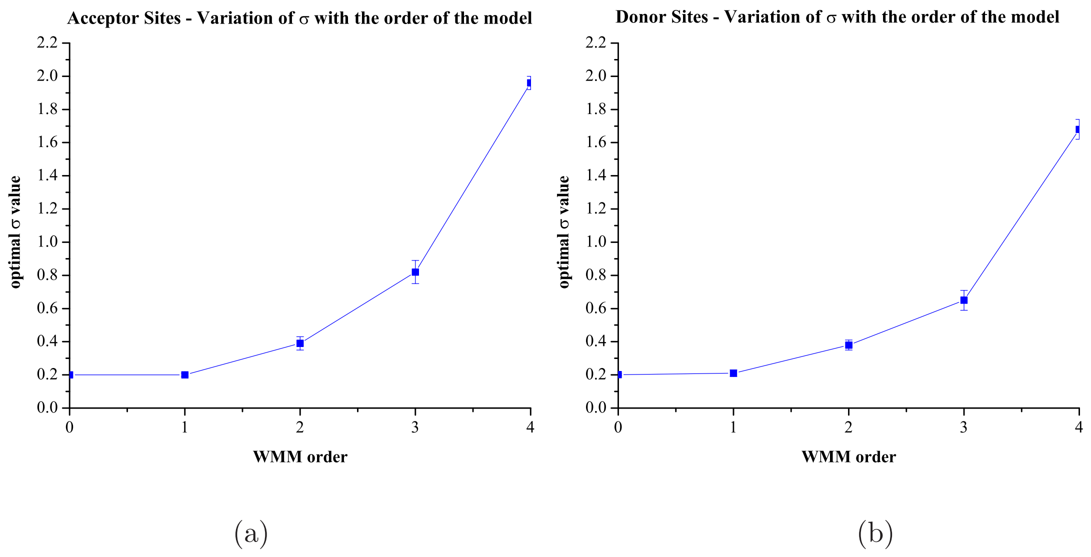


Figure 5.34: Variation of σ with the order of the Gaussian-smoothed weight matrix models: (a) Acceptor Sites, (b) Donor Sites.

5.5.3 Conclusions

This work shows that Gaussian-smoothed weight matrices are a better representation for acceptor site sequences compared to standard weight matrices, and also a good representation for donor sites. The smoothing provides a method to account for small sample size, avoiding the problem of having zero-probabilities that would dramatically affect the score of the signals; as our results show, Gaussian smoothing constitutes an alternative to the standard pseudo-count operator, and it becomes certainly more appropriate when there exist correlations between closely located signals in the sequence; standard pseudo-counts are quite arbitrary, as represent a weighted sum of the probabilities over the whole sequence, which is not an adequate assumption when the signals are independent.

Acceptor splice sites are better characterised by the presence of long k-mers, which accentuates the problem of the small sample size; the benefit of using Gaussian-smoothed probabilities is evidenced by the results. On the other hand, the performance of the optimal Gaussian-smoothed WMMs for donor sites is comparable to the one of the best standard WMMs; this is due to the fact that the best representation for donor sites seems to be based on short k-mers, where the smoothing does not provide any favourable position.

5.5.4 Integration of the Gaussian-smoothed weight-matrices into a gene prediction program

In order to evaluate the performance of our Gaussian-smoothed position-specific weight matrices in the context of gene prediction, we tested a simple integration with the program AUGUSTUS [Stanke and Waack, 2003, Stanke et al., 2004, Stanke and Morgenstern, 2005]. AUGUSTUS is a gene prediction program based on a generalised hidden Markov model (GHMM) that allows incorporating extrinsic information in the form of binary *hints*; this information can overlap with the information that the program itself is retrieving from the sequences, as it is the case with splice sites.

AUGUSTUS model

AUGUSTUS model consists on several states representing the different parts of a genomic sequence, which are thought to create nucleotide sequences with specific emission probabilities. Extrinsic hints have the effect of modifying the probability of the models; regarding splice sites, this is done according to the following rules:

- Gene structures that respect hints get a *bonus* with regard to those that ignore the hints.
- Insecure hints are not necessarily respected.
- If real genes are mainly supported by extrinsic information, gene structures for which there is no supporting extrinsic information get a *malus*.

The hints can be of different *types*, depending on the sequence section they refer to, and of different *grades* (i.e., different levels of reliability, sources, etc). Our hints are of two types, one for acceptor sites and the second one for donor sites, and consist of binary information about the presence or absence of an acceptor site or a donor site at a precise location. The influence of the hints depends on the ratio of hints that is compatible with real gene structures, which must be determined in advance on training data.

Hints have the effect of modifying the emission probabilities of AUGUSTUS original GHMM. Let $e_{p,q,\tau}(\sigma)$ be the probability of emitting a nucleotide sequence $\sigma = \sigma_1\sigma_2\dots\sigma_n$ from state q, given that the previous state is p, and the previously emitted nucleotide sequence is τ. If we now consider the hints, the probability of emitting the same nucleotide sequence is redefined as the probability of emitting the nucleotide sequence, $e_{p,q,\tau}(\sigma)$, multiplied by the probabilities of the hints given. In order to specify those probabilities, the hints must first be evaluated on training data for which the correct annotation is given. While hints that are compatible with real gene structures are *good hints*, hints that are not compatible with the annotation are *bad hints*.

Let $p_j^+(g)$ denote the probability that a respected hint of type j has grade g, and $p_j^-(g)$ the probability that a non-respected hint of type j has grade g. And let r_j^+ be the probability of observing a respected feature of type j, and r_j^- be the probability of observing a non-respected feature of type j in the training data. Then, the probability of observing a certain hint f at position k, $P(F_k = f)$, can be defined as:

$$P(F_k = f) = \begin{cases} p_j^+(g)r_j^+, & \text{if } f \text{ is respected by the structure;} \\ p_j^-(g)r_j^-, & \text{if } f \text{ is not respected by the structure, but still} \\ & \text{observable at that position;} \\ 0, & \text{if } f \text{ is not observable from a biological point of} \\ & \text{view at that position.} \end{cases} \tag{5.37}$$

The probability $p_j^+(g)$ can be estimated from the relative frequency of grade g among all good hints of type j in the training set. In a corresponding manner, $p_j^-(g)$ can be estimated from the relative frequency of grade g among all bad hints of type j in the training set. The probabilities r_j^+ and r_j^- can be estimated as:

$$r_j^+ = \frac{\#(good\ j\ hints)}{\#(j\ features)} \tag{5.38}$$

and

$$r_j^- = \frac{\#(bad\ j\ hints)}{\#(2 \cdot rate(j)L)} \tag{5.39}$$

where L is the length of the sequence.

When a structure respects a certain hint of type j and grade g, then its emission probability is multiplied by a *bonus.* The bonus for a hint of type j and grade g is defined as

$$bonus(g,j) = \frac{p_j^+(g) r_j^+}{p_j^-(g) r_j^-}. \tag{5.40}$$

On the contrary, the emission probability of structures that do not respect a given hint is penalised by a *malus.* Calculating the malus involves estimating the probability of not observing a particular hint at a position where it is observable from a biological point of view, $P(F = \Phi)$. In the case of acceptor sites, this is

$$P(F = \Phi) = \frac{\#(acceptor\ sites\ without\ hints)}{\#(acceptor\ sites)} \tag{5.41}$$

and for donor sites

$$P(F = \Phi) = \frac{\#(donor\ sites\ without\ hints)}{\#(donor\ sites)}. \tag{5.42}$$

Let F_s be a random hint f in a structure that supports or is compatible with feature f, and F_u be a random hint f in a structure that does not support hint f, then the *malus* can be defined as

$$malus(g,j) = \frac{P(F_s = \Phi)}{P(F_u = \Phi)}. \tag{5.43}$$

Methods

Datasets

The training sequences for these experiments were obtained from the set of 1284 single gene sequences from GenBank that was originally used to train AUGUSTUS. The test sequences are 178 human genomic sequences in GenBank flat file format with one complete gene each.

Experimental setup

The experiments involved the calculation of the *bonus* and *malus* for groups of splice site hints obtained using different criteria, and the testing of the performance of AUGUSTUS with the new parameters.

We evaluated each splice site in the datasets using Gaussian-smoothed position-specific weight matrix models, and computed a score. This score was used to sort the signals. We performed a series of experiments using different training set sizes, including 100, 200, 300, and 400 sequences, and observed the influence on the results on a separate testing set. We created hints using different strategies:

- using score thresholds to divide the signals into hints of two grades, based on our prior knowledge that low-score signals should include less real splice sites;
- using exclusively signals that are local maxima in terms of their scores along the sequence;
- using only the signals whose scores are absolute maxima in the given sequence.

This choice relies on the score distributions observed for acceptor and donor site signals, which are detailed below.

Score distribution of the splice site signals

Figure 5.35 corresponds to the score distributions produced by the Gaussian-smoothed position-specific weight matrix models on the dataset used for training in the experiments presented in the following sections. As expected, the distributions for real splice sites are notably shifted to higher scores, while the distributions for pseudo splice sites are shifted to lower scores. This explains the high AUC values obtained during the previous evaluation of the method, which were also reproduced on these sequences. However, the ratio of real splice sites to pseudo splice sites is very low, namely 0.6570% in the case of acceptor sites, and 0.9519% in the case of donor sites. Consequently, even if the distributions differ considerably, classifying the signals based on their score will result in a large absolute number of false positives.

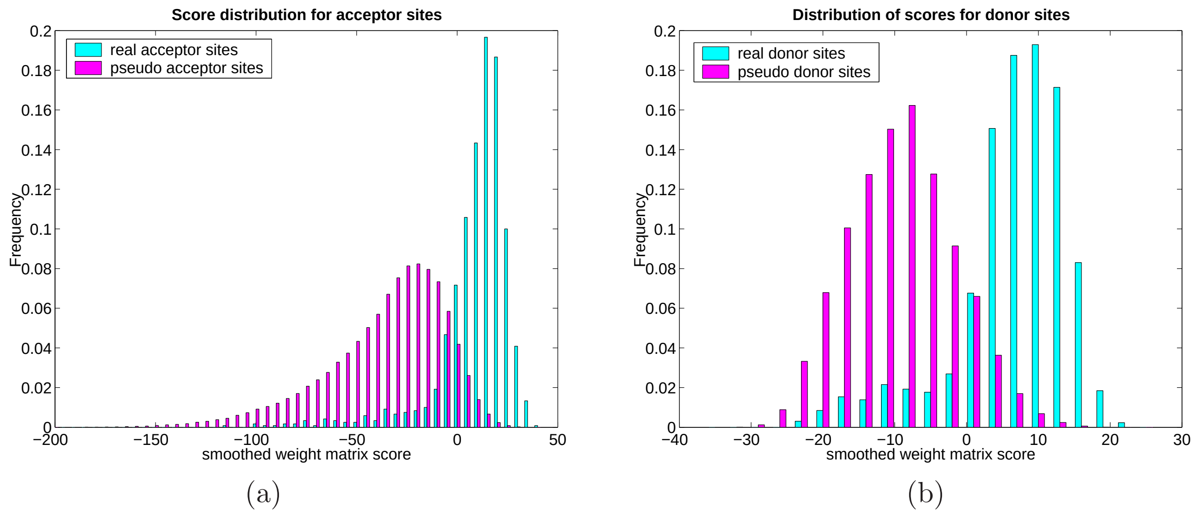


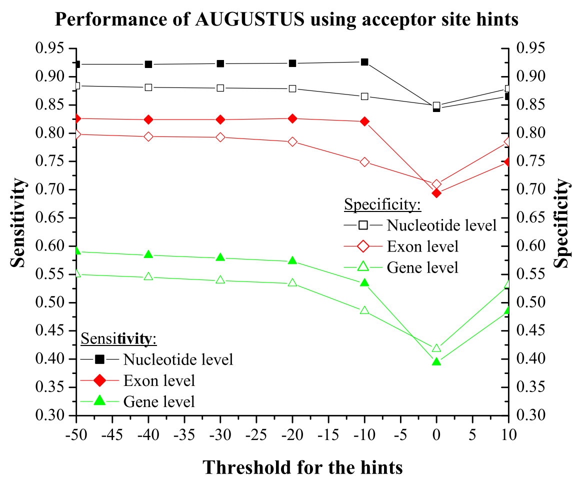
Figure 5.35: Distributions of the scores produced by the smoothed position-specific weight matrix models for acceptor sites (a) and donor sites (b).

In the case of donor sites, the mean across all sequences is of −8 and the standard deviation of the mean is 7. For acceptor sites, the mean is −34 and the standard deviation of 29.

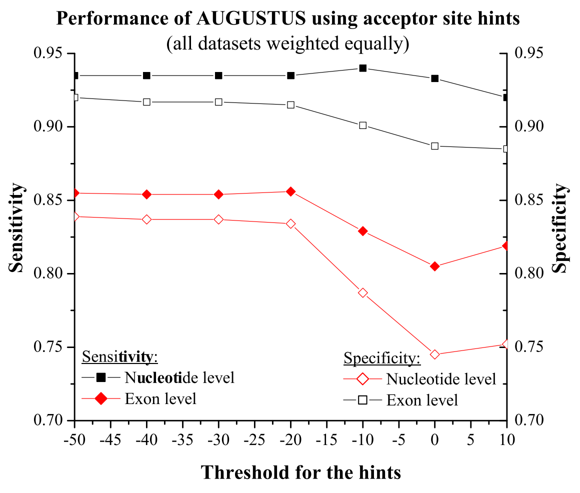
Results

The first experiment consisted in separating the signals scored by our Gaussian-smoothed position-specific weight matrices into two grades, one grade for high-score signals, and a second grade for low-score signals. This is based on the fact that the distributions of pseudo splice sites and real splice sites have different means. Acceptor splice sites were separated into two grades or groups, according to the thresholds $\{-50, -40, -30, -20, -10, 0, 10\}$, and donor splice sites according to $\{-10, -5, 0, 5, 10\}$. The results are depicted in Figure 5.36. Using −5 as a threshold to separate donor sites into two categories slightly improves the sensitivity at nucleotide level without changes on the specificity; however, the performance at the exon and gene level is negatively affected. If the test sequences are weighted equally, using donor sites as hints can improve the sensitivity and specificity of the program at nucleotide and exon level. On the contrary, using acceptor sites as hints does not seem to produce any advantages with respect to the original performance of the program.

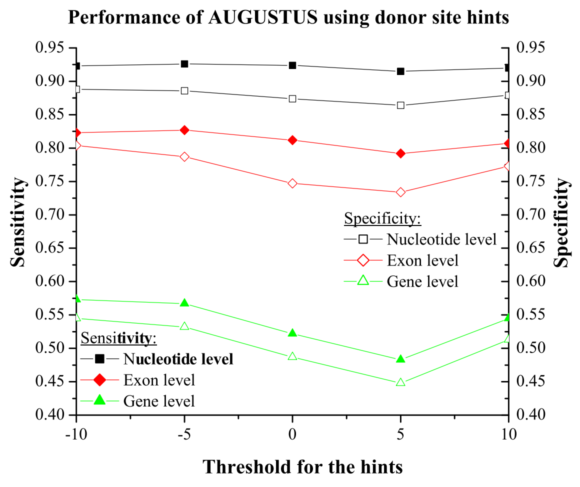
Hints corresponding to local and absolute maxima signals do not produce any significant modification on the performance of the program.



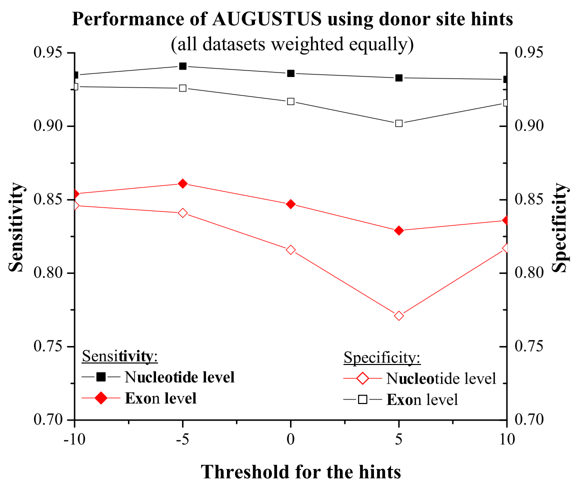
(a)



(b)



(c)



(d)

Figure 5.36: Performance of AUGUSTUS using two *grades* of acceptor (a, b) and donor (c, d) site hints; the grades were defined using score thresholds.

Discussion and conclusions

Under the conditions described here, the use of hints does not improve the performance of AUGUSTUS. This is due to several reasons. In the first place, the use of hints implies that the splicing windows are considered twice, once by AUGUSTUS and a second time in the form of the hints. The influence of the hints on the score of the predicted structures depends on the accuracy of the hints; due to the fact that pseudo splice sites occur in much greater number than real splice sites, it is unlikely to be able to provide enough good hints to improve the performance of AUGUSTUS without including a large number of bad hints. Moreover, if the ratio between good hints and bad hints is high, all hints will have a strong influence on the performance, including bad hints.

The results could change if AUGUSTUS was using exclusively the splice site prediction matrices we suggest. Indeed, AUGUSTUS uses a model similar to the ones that have been compared to ours. It uses monomer distributions for windows of k nucleotides centred at the splice site dimers, and higher order Markov chains to model the branch point and the coding and non coding regions. Our model has proved to be superior to 0th-order Markov chains, and to combinations high-order position-specific weight matrices, that should make it an attractive option for replacing its AUGUSTUS model. However, the improvement we have shown, even if significant, is slight. Replacing AUGUSTUS model with ours would imply reformulating the algorithm, as the different sequence regions would need to be adapted to the window sizes we employ, and would therefore be considered differently, requiring an extensive retuning of the program parameters.

Chapter 6

Intron Evaluation using Covariation Models

6.1 Introduction

Consistent covariation of two nucleotides in a Watson-Crick manner suggests some sort of relationship between those two positions in the secondary structure of RNA. The idea is to examine the RNA sequences looking for compensatory mutations; these are likely to indicate pairing bases in the RNA secondary structure. For instance, given an alignment of sequences, if positions i and j of the sequence correspond to pairing bases, a change from A to C at position i should be accompanied with a change from U (T in the DNA sequence) to G at position j.

RNA splicing is a process that removes introns and joins exons in a primary transcript. An intron usually contains a clear signal for splicing. The detailed RNA splicing mechanism is not simple; it involves five snRNAs and their associated proteins, which together form a large (60S) complex, called spliceosome. The intron is removed after a two-step enzymatic reaction, and the two neighbouring exons are joined together; the branch point A residue plays a critical role in the enzymatic reaction. This mechanism suggests the requirement for a particular spacial arrangement of the RNA molecule.

Hereby, we propose the use of RNA secondary structure homology as a method to identify introns.

6.2 Method

6.2.1 Model

Our intron prediction method relies on covariation models, which we estimate by means of the tool COVE v. 2.4.4, by Eddy and Durbin [1994]; this is a program designed to analyse RNA structure by means of covariation models.

COVE calculates a multiple sequence alignment on the training sequences, which is not based on conserved positions, but on covarying positions, as described in section 2.6. The consensus derived from the resulting alignment is a secondary structure model, which can be compared to a new set of sequences and used to score them depending on the shared homology. Sequences with a covariation pattern similar to the one of the model will receive positive scores, and sequences not matching the model pattern will receive negative scores.

COVE calculates the model using an iterative process. In the initial step, COVE calculates a random multiple alignment of the training sequences and its corresponding model, which will serve as starting point for the first iteration. Next, the program refines the alignment by identifying correlations and conserved positions, and this process is repeated until neither the model structure nor its parameters are changing significantly.

We show how to use COVE to distinguish between real introns and false (pseudo) introns in an RNA sequence. We employ COVE to obtain a model based on the alignment of sequences containing a donor site on the 5' end and the corresponding acceptor site on the 3' end, and then evaluate the homology of further sequences; sequences sharing a high homology with the model are classified as *real introns*, while sequences sharing a low homology with the model are classified as *pseudo introns*.

6.2.2 Datasets

For our tests, we employed the following data sets:

- HS3D (*Homo sapiens* Splice Sites Dataset).
- A set of 117 pairs of genomic sequences from human and mouse compiled by Batzoglou et al. [2000].

The first data set was used for creating our intron model, and for general performance tests. The second data set was only employed for testing our intron model on more realistic data; the results of this analysis are reported in section 6.7.

6.2.3 Sequence representation

In the case of the HS3 dataset, which consists actually on two datasets, one for acceptor splice sites and one for donor splice sites, extracting intron sequences implied joining the corresponding donor and acceptor sites from the two different datasets; the procedure is possible for sequences that overlap, because the sequences are labelled, but leads to some limitations; as the donor and acceptor site sequences in this dataset have fixed lengths, the length of the resulting intron sequences is also restricted. After the necessary pre-processing of the data, we extracted intron sequences with a series of properties.

The sequences collected by Batzoglou et al. [2000] are human and murine genomic sequences, each containing at least one complete gene structure; the extraction of intron sequences relying on the available annotations is in this case straightforward. To ensure the homogeneity of the training and test sets we utilised only human sequences.
Sequence format example:
>Sequence_Name
5'-ATGGAGATGACAGAGGCTGCCCTGCGCCTGCTGAGCAGGAACCCCCGCGGCTTCTACCTCTTTGTGGAGG|*GTG*
CGTGGTGGCCCCTGGGGAGTGGAGGAAGGCGGGGCGCGGCAGGGCAGGTTCAAGCATCACCCCCCTCTGGCCTTCCT
GCAG|GCGGCCGCATCGACCATGGTCATCATGAGGGTGTGGCTTACCAGGCACTCACTGAGGCGGTCATGTTCGA-3'

- The sequences consist of 70 nucleotides before the occurrence of the relevant GT (GU in the pre-mRNA sequence), 70 nucleotides after the relevant AG, and the complete nucleotide sequence in between the splice sites.

- If the GT corresponds to a real donor site and the AG corresponds to a real acceptor site, the sequence is labelled as *true intron* or simply *intron* (positive example); otherwise, the sequence is labelled as *pseudo intron* (negative example).
- The symbol "|" represents the borders of the intron; the intron sequence is written in italics.

6.2.4 Classification criteria

The performance of the classifier was evaluated in terms of classification rate (CR) and the area under the ROC curve (AUC).

6.3 Running time tests

Programs based on covariation models have serious computational limitations; these programs are normally slow and employ a lot of memory, especially during the training phase. Both memory and time requirements increase rapidly with the number and individual lengths of the sequences involved in the training. Table 6.1 shows how the run time of COVE increases exponentially with the number of sequences used for training; this is a severe constraint for producing a powerful model.

Number of sequences	Running time [min]	Iterations	Model
4	1.90	2	1
8	6.28	3	2
16	32.13	5	3
32	2296.77	100	4
64	627.82	9	5
128	15388.05	79	6
256	5350.58	9	7
264	23963.28	39	8

Table 6.1: Run times of COVE depending on the number of training sequences.

6.4 Results

6.4.1 Performance of the method

We first tested the effect of the number of training sequences on the prediction power of the model. For this purpose, we created different models with an increasing number of positive examples, i.e., intron sequences; in contrast, all models were tested on exactly the same sequence set, formed by 251 positive examples (intron sequences) and 258 negative examples (pseudo introns).

As it can be observed in Table 6.2 and Figure 6.1, the prediction power of the model strongly depends on the amount of sequences that was used to generate it. Models created with 4 and 8 sequences exhibit a limited performance; nevertheless, in spite of the low number of sequences involved in the training, the performance of the models is clearly not random, suggesting that they are grounded on some biological property of the sequences. The model which shows the best performance (*Model 8*) coincides with the model trained with the highest number of sequences (264).

Model	Training Set	CR [%]	AUC [%]
1	4	0.71513	0.77739
2	8	0.7112	0.77827
3	16	0.73674	0.8172
4	32	0.77407	0.84539
5	64	0.81532	0.8901
6	128	0.86248	0.93435
7	256	0.88605	0.95165
8	264	0.89587	0.95587

Table 6.2: Influence of the training set size on the test performance of the covariation model. The column *Training Set* shows the number of sequences used for generating the model.

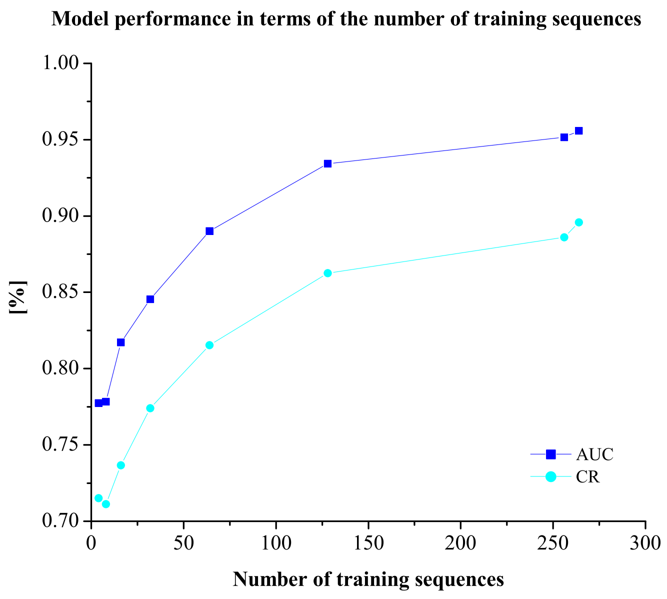


Figure 6.1: Intron prediction using covariation models. Performance of the method as a function of the number of sequences used for training the model.

6.4.2 Underlying alignment

The secondary structure models produced by COVE rely on the multiple alignments of the training sequences. In the case of introns, the most trivial positions expected to be aligned are the occurrences of the donor site, GT (GU in the pre-mRNA sequence), and the acceptor site, AG. As a matter of fact, this was observed with a few exceptions on all resulting multiple alignments, independently of the number of sequences involved in the training. For that reason, we also explored the influence of the misaligned splice sites on the general performance of the method; these experiments were executed on the models created with 64 and 264 sequences (model 5 and 8, respectively). As COVE does not provide any means to force the alignment of the sequences at certain positions, we opted for eliminating from the original training sets the sequences whose acceptor or donor sites were misaligned with respect to the rest of the sequences, and created new multiple alignment and models with these *curated* sequence sets.

The multiple alignment corresponding to *model 5* contains 16 misaligned sequences; from these sequences, four have both splice sites misaligned, eight have their donor sites misaligned, and four have their acceptor sites misaligned. In the case of *model 8*, from 13 misaligned sequences, three have their donor sites incorrectly aligned and ten have their acceptor sites incorrectly aligned; there are no sequences with both splice sites misaligned. The models generated with the *curated* training sets were used to construct new models, which were in turn evaluated on the same test set used to evaluate the former models.

The performance of the new models, *model 5'* and *model 8'* (shown in Table 6.3) is inferior to the one of the original models. Regarding the alignment corresponding to *model 5'*, 47 sequences have their donor sites correctly aligned, but the acceptor sites are aligned within a range of 59 positions; one sequence is not aligned with the rest. On the other hand, the alignment corresponding to *model 8'* contains ten incorrectly aligned sequences (three whose donor sites are misaligned, and seven whose acceptor sites are misaligned). We conclude that this manual correction does not improve the predictive performance of the method.

Model	Training Set	CR [%]	AUC [%]
5'	48	0.78782	0.88285
8'	251	0.88212	0.94943

Table 6.3: Performance of the models generated with the training sets of models 5 and 8 after removing all misaligned sequences in the original alignments. The column *Training Set* shows the number of sequences used for generating the model.

6.5 Reference method for comparison

We compared our results with the performance of position-specific weight matrices of order 0, 1 and 2. The comparison is not straightforward; while we use the covariation models to classify introns, position-specific weight matrices predict splice sites.

The tests were done using as parameter the window over the sequence for which the weight matrices were calculated; for a model of order k, the left and right ends of the window were chosen from the set $\{-50, -40, -30, \ldots, +40, +50 - k\}$; the position where the first letter of the consensus dinucleotide (AG in the case of acceptor splice sites, GT in the case of donor sites) occurs corresponds to 0; upstream positions are represented by negative positions, and downstream positions by positive positions. The experiments were run on the HS3D sequences, using a 3-fold cross-validation scheme, with an internal 5-fold cross-validation scheme designed to select the optimal window; the sets contained an equal number of positive and negative examples (real and pseudo acceptor or donor sites). We computed a model for real splice sites (positive class) and another model for pseudo splice sites (negative class), and classify the test examples according to their probability of belonging to either of the classes.

The best average models are presented in Table 6.4. The performance of the covariation models applied to the prediction of introns are slightly inferior, although we cannot compare their significance. In addition, correctly predicting an intron is equivalent to correctly predicting its donor site and its acceptor site; if we assumed that such predictions are independent (which of course is not absolutely true), then we could estimate that the AUC values for the prediction of introns using position specific weight matrices would be actually inferior to the ones obtained with the covariation methods. However, as stated before, a direct comparison in this case is not appropriate, and is only meant to establish the validity of our approach.

Signal	order	Window	CR	AUC
Acceptor sites	2	[-17 ± 1, 1 ± 1]	0.894 ± 0.005	0.954 ± 0.001
Donor sites	2	[-6 ± 0, 10 ± 0]	0.92 ± 0.01	0.967± 0.008

Table 6.4: Splice site prediction using position-specific weight matrix models (reference).

6.6 Intron Model

We explored the multiple alignment used to generate the best covariation model, *model 8*, looking for secondary structure patterns. We found six pairs of nucleotides with a presumed role in secondary structure, whose absolute positions in the sequence are detailed in Table 6.5. The positions indicated in the table are with regard to the donor site (DS) and to the acceptor site (AS) and independent of the number of gaps added for constructing the alignment; the sequences have an average length of 243 nucleotides. We indicate the range of positions in which the covarying nucleotides are located, depending on the particular sequence. The mutual information is calculated according to Equation 2.46; in the case of donor and acceptor sites (in Table 6.5, covarying nucleotides 2 and 6, respectively), the values are due to one and four clear misalignments.

Covarying	Position relative to DS				Position relative to AS				Mutual
nucleotides	Range	Mean	Range	Mean	Range	Mean	Range	Mean	inf. [bits]
1	[-44, 2]	-44	[4, 51]	5	[-114, -176]	-126	[-66, -128]	-78	0.0883
2	[0, 46]	0	[1, 47]	1	[-70, -132]	-92	[-69, -131]	-81	0.0359
3	[35, 90]	46	[79, 141]	110	[-33, -93]	-42	[9, -40]	9	0.4290
4	[43, 101]	59	[45, 104]	64	[-25, -84]	-31	[-23, -79]	-28	0.8696
5	[65, 127]	94	[72, 134]	103	[-3, -54]	-5	[2, -47]	2	0.2490
6	[70, 222]	101	[71, 133]	102	[0, -49]	0	[1, -48]	1	0.0926

Table 6.5: Intron prediction using covariation models. Information about the position of the covarying nucleotide pairs found relative to the donor site (DS) and acceptor site (AS) of the intron sequence.

The donor and the acceptor sites are absolutely conserved regarding the primary sequence (see Figures 6.2 and 6.5). For the first and third covarying pair, there is no obvious consensus in the sequence, and the interpretation is difficult. Similarly, the neighbouring bases of the remaining covarying nucleotides (see Figures 6.3 and 6.4) do not exhibit high conservation. As a conclusion, with exception of the splice sites, there is no motif related to any of the covarying pairs, which may explain the covariation observed.

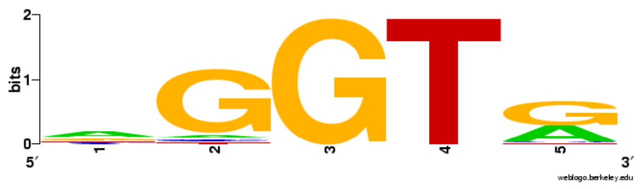
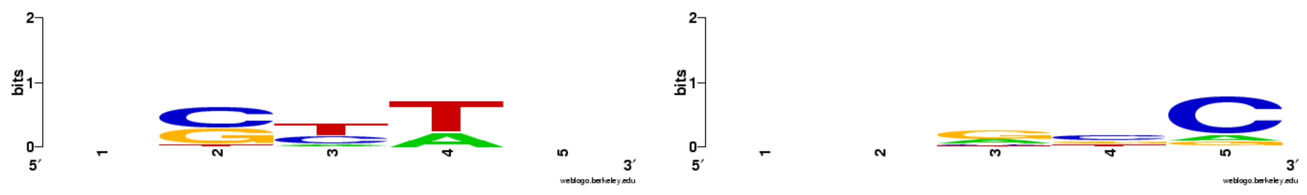
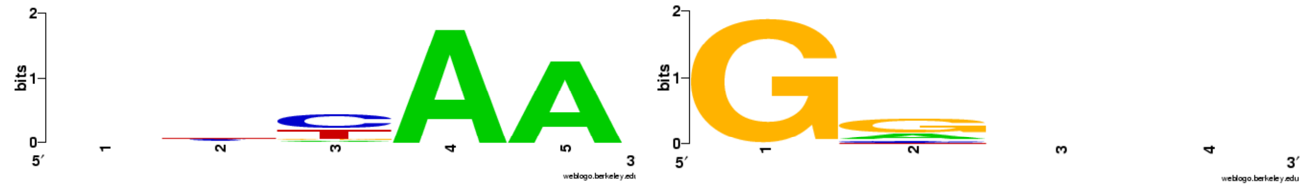


Figure 6.2: Donor splice site or *covarying nucleotides 2* (positions 3 and 4) consensus.



(a) (b)

Figure 6.3: Motifs corresponding to *covarying nucleotides 4* (the covarying nucleotides can be seen at positions 3 and 3, respectively).



(a) (b)

Figure 6.4: Motifs corresponding to *covarying nucleotides 5* (the covarying nucleotides can be seen at positions 3 and 2, respectively).

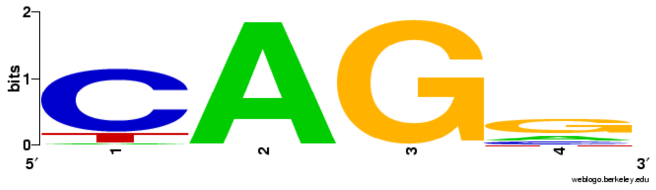
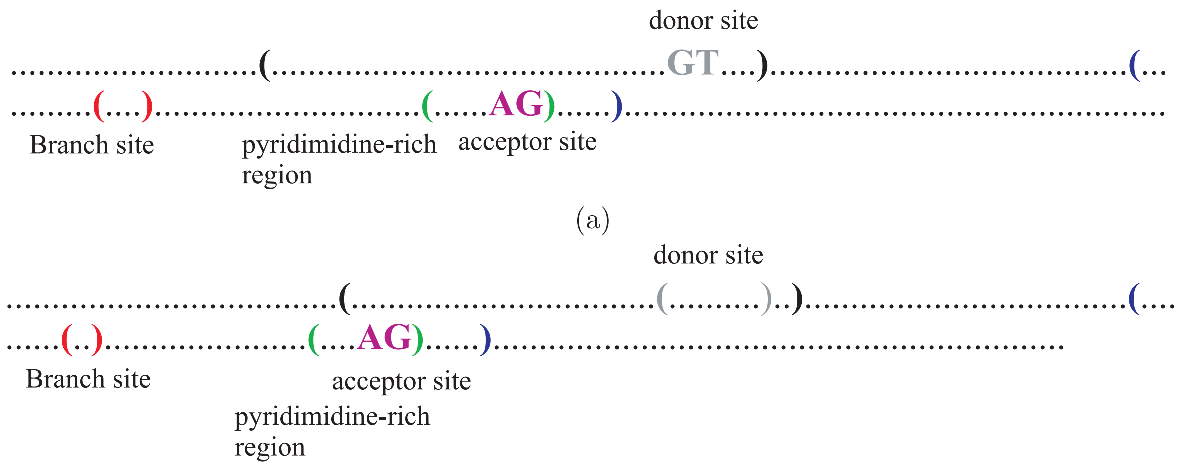


Figure 6.5: Acceptor splice site or *covarying nucleotides 6* (positions 2 and 3) consensus.

In spite of demonstrating a good performance, the model only includes a few pairs of covarying nucleotides (Figures 6.6), a finding which is not consistent with strong structural conservation. We might then conjecture that there are different groups of introns whose members share structural properties, and that the model obtained here captures only the characteristics common to all groups.


donor site
...........................(..GT....)..(...
.........(....)............................(......AG)......)..
Branch site
pyridimidine-rich region
acceptor site
(a)
donor site
...................................(.................................(..........)..)...................................(....
......(..).....................(....AG)......)..
Branch site
acceptor site
pyridimidine-rich region

(b)

Figure 6.6: Intron prediction using covariation models. General intron model based on the location of covarying nucleotides, relative to the donor (a) and to the acceptor (b) splice sites.

6.7 Final evaluation

After calculating and optimising the best model (*Model 8*) using the sequences in HS3D, we tested it on human sequences extracted from the data set compiled by Batzoglou et al. [2000]. In contrast to the sequences in HS3D, the sequences collected by Batzoglou et al. [2000] are human and murine genomic sequences that contain at least one gene, with all its constituting exons and introns. HS3D is actually an acceptor site data set and a donor site data set, both composed by 140 nucleotide-long sequences; in order to obtain the intron sequences employed for the experiments presented in this work, we have pre-processed the sequences, by joining the donor splice site sequences with the corresponding acceptor splice site sequences; this pre-processing step requires the existence of an overlapping region between the donor site and the acceptor site sequences, which limits the length of the resulting intron sequences. Therefore, the database proposed by Batzoglou et al. [2000] constitutes a more natural test scenario for our model. To avoid memory problems while running COVE, we limited the length of the sequences extracted from this data set to a maximum of 500 nucleotides; the minimal length was 30 nucleotides.

6.7.1 Performance of the method on general data

Firstly, we evaluated *model 8* on a set of 238 positive examples and 214 negative examples extracted from the database of Batzoglou et al. [2000]; the negative examples were selected randomly. These tests produced a ROC curve area (AUC) of 0.94943 and a classification rate (CR) of 0.88212, values comparable to the ones obtained for the previous evaluation of the method on HS3D.

6.7.2 Performance of the method on data including only *proximal* pseudo introns

This last test was designed to evaluate the performance of the covariation models for the prediction of introns under stringent conditions. The most interesting problem in the context of gene prediction consists on distinguishing real splice sites from pseudo splice sites that are located in close proximity to the formers. For the purpose of this analysis, we selected pseudo introns whose boundaries were in a range of ±25 nucleotides with respect to the location of the real splice sites; we refer to these pseudo introns as *proximal* negative examples.

We tested the model that was found to perform best in experiments described above on a set composed by 238 positive examples (introns) and 245 proximal negative examples (pseudo introns). These tests produced a ROC curve area (AUC) of 0.68067 and a classification rate (CR) of 0.65424. Figure 6.7 shows a comparison between the ROC curve of the method tested on randomly chosen negative examples and the ROC curve of the method tested on *proximal* negative examples; the performance of the method on data containing exclusively *proximal* negative examples, although not random, is significantly worse. However, these results were expected considering that the model is mainly based on only six pairs of nucleotides.

6.8 Conclusions

The method performs very well on random data, but its performance is poor when evaluated on pseudo introns that overlap with real introns; this is a reasonable and even expected outcome, considering that covariation models identify analogous structure patterns, and that pseudo introns overlapping with real introns should share the characteristic structure patterns.

Anyway, the method might definitely be useful to quickly scan intron candidates and discard

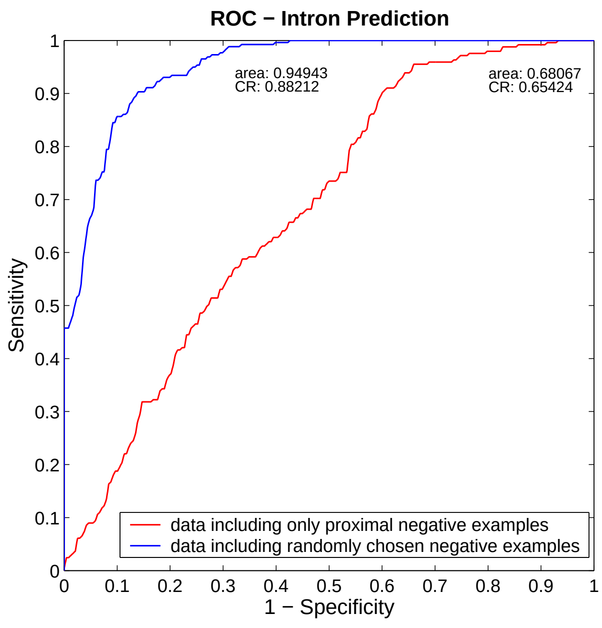


Figure 6.7: Intron prediction using covariation models. Final evaluation.

or downweight those that receive a poor score, as it has proved to distinguish quite well between random data and sequences that are structurally related to introns. In addition, it could be combined with a method able to recognise, for example, the consensus regions of donor and acceptor sites, which should compensate the aforementioned limitations.

Chapter 7

Summary and Conclusion

7.1 Summary of results

Splice site prediction is the most decisive task for the accuracy of eukaryotic gene prediction. In the context of gene prediction, this thesis has presented a detailed study of the splice site prediction problem, which has the potential of improving the design of further systems. The results can be divided into four parts; we have discussed the implementation of a web server for the gene prediction program AGenDA [Rinner and Morgenstern, 2002] which simplifies the usage of the software and makes it publicly available for the scientific community; then, we have introduced two different approaches to the splice site prediction problem itself, one which focuses on the understanding of the biological mechanism of splicing, and one which represents an enhancement of existing prediction systems; finally, we have evaluated an intron prediction system based on secondary structure properties of RNA sequences. A brief review follows.

Firstly, we have introduced a web server for the gene prediction program AGenDA [Rinner and Morgenstern, 2002]. AGenDA is a gene prediction software that assumes that sequences that are conserved between closely related species are likely to be functional. Functional regions are subject to evolutionary pressure, and thus, less variable than non-functional regions. Such conserved regions are identified using alignment programs. The web server automatises the processes that create the input for the program, and produces an informative output for the user. In addition, we have improved the algorithm that retrieves the optimal gene structure from the output list of exons produced by AGenDA.

Regarding splice site classification, we have described two new approaches, the first one based on Support Vector Machines (SVMs) and oriented towards the recognition of biologically meaningful patterns, and the second one based on position-specific weight matrix models (WMMs). SVMs are machine learning systems that deduce a classification function or *rule* from a training data set. By using an appropriate representation for the sequences, we have shown how to obtain information about the biological processes involved in the mechanism of splicing, in the form of specific patterns, from the discrimination function of such models. Our study exemplifies how to profit from such a learning technique, not only as a prediction tool, but also as a source of potentially biologically meaningful data. The second contribution of this thesis regarding splice site prediction is an heuristic method to improve the performance of position-specific weight matrices (WMMs). In this case, we have shown how to overcome problems associated with the limitation of available data for the construction of high-order models, by applying of a Gaussian smoothing filter. Our model has shown to perform very well on the splice site prediction problem, and could be integrated in the future into new gene prediction programs.

At last, we have demonstrated that secondary structure properties, in the mathematical description given by covariation models, can become a valuable tool for developing intron prediction systems. Covariation models identify pairing nucleotides in RNA sequences, i.e., the units responsible for the secondary structure of the molecule; this is accomplished by assuming that functional RNA secondary structure is highly conserved, which requires that mutations in one of the nucleotides involved in such pairs are immediately compensated by another mutation in the corresponding pairing nucleotide [Hofacker et al., 2002]. Using available software [Eddy and Durbin, 1994], we have shown how to obtain a model that, in spite of not offering much information about the secondary structure of introns, is clearly capable to distinguish introns from random sequences.

7.2 Contribution of the work

Compared to previous work, our approach to splice site prediction with SVMs is a more exhaustive study of the problem; we have represented sequences by the occurrences of substrings of lengths one to six with an adaptable degree of position-invariant information, which makes possible the identification of patterns that would be normally overseen or, at most, mere conjectures. We have accompanied our analysis with an intuitive visualisation of the decision function of the SVMs, which provides a deep insight into the patterns that characterise the sequences. We have confirmed the information gained by this means by comparing it to the output of standard feature selection methods and simple data statistics. As a conclusion, we have offered a more comprehensive, elaborated and systematic study of the nucleotide composition of splice site sequences. According to our findings, acceptor site sequences are characterised by position-dependent nucleotides in a short consensus region ligated to the highly conserved dinucletide AG, which is immediately preceded by a pyrimidine-rich (cytidine and thymidine) region, and the occurrence of a short region mainly composed by thymidine located upstream of the conserved AG but whose exact position varies; referring to donor site sequences, we show that they are characterised by highly correlated position-dependent nucleotide occurrences in a relatively long consensus region including the highly conserved dinucleotide GT, and regions with a specially low frequency of thymidine upstream of the conserved GT.

Position-specific weight matrices and Markov chains are one of the methods of choice for splice site prediction (see Rajapakse and Ho [2005], Cai et al. [2000]). The method that we have presented here represents an improvement for higher-order positional weight matrices, which are the models exhibiting the best performance. Higher-order models are able to capture complex interactions between nucleotides in the sequence, but are associated with a number of limitations, like high state-space complexity and reduced coverage. We have overcome the problems originated by small sample sizes based on the assumption that probabilities of adjacent positions in the sequence are not completely independent; by means of a Gaussian smoothing operator, we have estimated the probability of observing a k-mer at a certain position conditioned on its presence or absence on the neighbouring positions. We have shown that *smoothed* weight position-specific matrix models constitute very good representations for acceptor and donor site sequences, compared to standard position-specific weight matrices.

The last contribution of this thesis concerns our intron prediction system. The design of an intron model relying on RNA secondary structure was motivated by the knowledge that RNA structure is frequently highly conserved, and that the interaction of the RNA sequence with protein complexes during the splicing mechanism suggests the existence of structural requirements. In spite of the weak evidence for structural conservation found, our method performs very well on random data; however, its performance is poor when evaluated on pseudo introns that overlap with real intron sequences; pseudo introns overlapping with real introns should actually share most characteristic structural patterns with real introns, which explains the outcome. However, such a method could possibly be combined with a method able to recognise, for example, the consensus regions of donor and acceptor sites, which should compensate this limitation.

7.3 Further work

Further work should be done for the integration of the methods described in this thesis, which are certainly valuable for the improvement of gene prediction programs. Patterns whose position is not fixed in the sequence are generally not appropriately captured by any splice site prediction method; however, especially in the case of acceptor sites, such features exhibit a very high prediction power, and should not be ignored.

We have proposed and implemented an integration between our position-specific weight matrix models and the gene prediction program AUGUSTUS [Stanke and Waack, 2003, Stanke et al., 2004, Stanke and Morgenstern, 2005]. We have classified splice sites as good or bad *hints* according to the score they were given by our model; AUGUSTUS modifies the scores of predicted gene structures according to hints, whose accuracy must be previously determined on a training set. Unfortunately, we were not able to reflect the good performance of our models on the overall performance of AUGUSTUS. One reason for this could be that the information provided by the hints overlap, in this case, with AUGUSTUS' original splice site models; AUGUSTUS' splice site model describes nucleotide distributions for windows of length k centred at the splice site, and higher order Markov chains to model the branch point and the coding and non coding regions. In addition, pseudo splice sites are much more numerous than real splice sites; as hints consist only of binary information, providing a number of good hints that would be large enough to improve AUGUSTUS' performance becomes very unlikely without simultaneously including many bad hints; furthermore, ensuring that only a few bad hints are included is counterproductive, as then all hints will have a very strong influence on the predicted structure, including the few bad hints. An alternative approach would be replacing AUGUSTUS' splice site system by our smoothed position-specific weight matrices. Nevertheless, replacing AUGUSTUS' model by ours would require extensive tuning, as the models describe different sections of the sequence and are not exactly exchangeable in terms of their output scores.

Splice site prediction is perhaps the most critical component of eukaryotic gene prediction programs; from that point of view, we may conclude that it would be logical to found the design of new gene prediction programs on the kind of information presented in this thesis. For example, we have detected as the most characteristic feature for acceptor site sequences a short and dense regular succession of thymidines whose location exhibits some variability in the upstream region; in addition, our results have shown that the pyrimidine-rich region that precedes acceptor sites plays a very important role in the classification. Because such patterns do not occur at fixed positions in the sequence, standard splice site prediction methods frequently undervalue or fail to identify them. With regard to donor sites, our results suggests that prediction methods should rely on the consensus region, but considering the existence of correlations between the nucleotides, as in maximal dependence decomposition models (MMD).

Bibliography

B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. D. Watson. *Molecular biology of the cell.* Garland Science, Fourth edition, 2002.

M. Arita, K. Tsuda, and K. Asai. Modeling splicing sites with pairwise correlations. *Bioinformatics*, 18 Suppl 2:27–34, 2002.

S. Degroeveand B. De Baets, Y. Van De Peer, and P. Rouzé. Feature subset selection for splice site prediction. *Bioinformatics*, 18 Suppl 2:75–75, October 2002.

S. Batzoglou, L. Pachter, J. P. Mesirov, B. Berger, and E. S. Lander. Human and mouse gene structure: Comparative analysis and application to exon prediction. *Genome Research*, 10:950–958, 2000.

M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, Second edition, 1990.

M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Linear Programming and Network Flows*. John Wiley & Sons, Second edition, 1993.

S. M. Berget, C. Moore, and P. A. Sharp. Spliced segments at the 5' terminus of adenovirus 2 late mRNA. *Proceedings of the National Academy of Sciences of the United States of America*, 74(8): 3171–3175, August 1977.

E. Birney and R. Durbin. Dynamite: a flexible code generating language for dynamic programming methods used in sequence comparison. *Proceedings International Conference on Intelligent Systems for Molecular Biology, ISMB, International Conference on Intelligent Systems for Molecular Biology*, 5:56–64, 1997.

E. Birney and R. Durbin. Using GeneWise in the Drosophila annotation experiment. *Genome Research*, 10:547–548, 2000.

E. Birney, M. Clamp, and R. Durbin. GeneWise and Genomewise. *Genome Research*, 14(5):988–995, May 2004.

C. C. Blake. Do genes-in-pieces imply proteins-in-pieces? *Nature*, 273(5660):267–267, May 1978.

M. Borodovsky and A. Lukashin. Eukaryotic genemark.hmm. (unpublished).

D. A. Brow. Allosteric cascade of spliceosome activation. *Annual Review of Genetics*, 36:333–360, 2002.

M. Brudno, M. Chapman, B. Göttgens, S. Batzoglou, and B. Morgenstern. Fast and sensitive multiple alignment of large genomic sequences. *BMC Bioinformatics*, 4:66, December 2003.

C. Burge and S. Karlin. Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology*, 268(1):78–94, April 1997.

C. B. Burge and S. Karlin. Finding the genes in genomic DNA. *Current Opinion in Structural Biology*, 8(3):346–354, June 1998.

M. Burset and R. Guigó. Evaluation of gene structure prediction programs. *Genomics*, 34:353–367, 1996.

D. Cai, A. Delcher, B. Kao, and S. Kasif. Modeling splice sites with bayes networks. *Bioinformatics*, 16(2):152–158, February 2000.

T. Cavalier-Smith. Selfish DNA and the origin of introns. *Nature*, 315(6017):283–284, May 23-29 1985.

T. Cavalier-Smith. Intron phylogeny: a new hypothesis. *Trends in Genetics*, 7(5):145–148, May 1991.

T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* The MIT Press and McGraw-Hill, First edition, 1990.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines (and other kernel-based learning methods).* Cambridge University Press, 2000.

L. Croft, S. Schandorff, F. Clark, K. Burrage, P. Arctander, and J. S. Mattick. ISIS, the intron information system, reveals the high frequency of alternative splicing in the human genome. *Nature Genetics*, 24(4):340–341, April 2000.

S. J. de Souza, M. Long, R. J. Klein, S. Roy, S. Lin, and W. Gilbert. Toward a resolution of the introns early/late debate: only phase zero introns are correlated with the structure of ancient proteins. *Proceedings of the National Academy of Sciences of the United States of America*, 95(9): 5094–5099, April 1998.

P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.

S. Dong and D. B. Searls. Gene structure prediction by linguistic methods. *Genomics*, 23(3):540–551, October 1994.

R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids.* Cambridge University Press, 1998.

S. R. Eddy and R. Durbin. RNA sequence analysis using covariance models. *Nucleic Acids Research*, 22(11):2079–2088, June 1994.

G. Fung and O. L. Mangasarian. Proximal support vector machine classifiers. In *Proceedings of the 7th ACM Conference on Knowledge Discovery and Data Mining*, pages 77–86. ACM press, 2001.

T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, N. Schummer, and D. Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, October 2000.

M. S. Gelfand, A. A. Mironov, and P. A. Pevzner. Gene recognition via spliced sequence alignment. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 93, pages 9061–9066, August 1996.

W. Gilbert. Why genes in pieces? *Nature*, 271(5645):501, February 1978.

W. Gilbert. The exon theory of genes. *Cold Spring Harbor symposia on quantitative biology*, 52: 901–905, 1987.

M. R. Green. Biochemical mechanisms of constitutive and regulated pre-mRNA splicing. *Annual Review of Cell Biology*, 7:559–599, 1991.

R. Guigó, S. Knudsen, N. Drake, and T. Smith. Prediction of gene structure. *Journal of Molecular Biology*, 226(1):141–157, July 1992.

R. Guigó, P. Agarwal, J. F. Abril, M. Burset, and J. W. Fickett. An assessment of gene prediction accuracy in large DNA sequences. *Genome Research*, 10(10):1631–1642, October 2000.

D. Gusfield. *Algorithms on strings, trees and sequences, Computer Science and Computational Biology.* Cambridge University Press, 1997.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

M. L. Hastings and A. R. Krainer. Pre-mRNA splicing in the new millennium. *Current Opinion in Cell Biology*, 13(3):302–309, June 2001.

I. Hofacker, M. Fekete, and P. Stadler. Secondary structure prediction for aligned RNA sequences. *Journal of Molecular Biology*, 319(5):1059–1066, 2002.

A. J. Horrevoets, G. Tans, A. E. Smilde, A. J. van Zonneveld, and H. Pannekoek. Thrombin-variable region 1 (VR1). Evidence for the dominant contribution of vr1 of serine proteases to their interaction with plasminogen activator inhibitor 1. *Journal of Biological Chemistry*, 268(2):779–782, January 1993.

M. Jacob and H. Gallinaro. The 5' splice site: phylogenetic evolution and variable geometry of association with U1RNA. *Nucleic Acids Research*, 17(6):2159–2180, March 1989.

T. Joachims. *Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning*, chapter 11. MIT-Press, 1999.

I. Korf. Gene finding in novel genomes. *BMC Bioinformatics*, 5:59, May 2004.

A. Krogh. Two methods for improving performance of an HMM and their application for gene finding. *Proceedings International Conference on Intelligent Systems for Molecular Biology, ISMB, International Conference on Intelligent Systems for Molecular Biology*, 5:179–186, 1997.

D. Kulp, D. Haussler, M. G. Reese, and F. H. Eeckman. A generalized hidden markov model for the recognition of human genes in DNA. *Proceedings International Conference on Intelligent Systems for Molecular Biology, ISMB, International Conference on Intelligent Systems for Molecular Biology*, 4:134–142, 1996.

Y. J. Lee and O. L. Mangasarian. RSVM: Reduced support vector machines. *Technical Report 00-07, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, July 2000. Proceedings of the First SIAM International Conference on Data Mining, Chicago, April 5-7, 2001, CD-ROM Proceedings*, 2000-2001.

C. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: a string kernel for SVM protein classification. *Pacific Symposium on Biocomputing, Pacific Symposium on Biocomputing*, pages 564–575, 2002.

H. F. Lodish, A. Berk, L. Zipurskyand, P. Matsudaira, D. Baltimore, and D. Darnell. *Molecular Cell Biology.* W. H. Freeman and Company, Fourth edition, 2000.

J. M. Logsdon, M. G. Tyshenko, C. Dixon, J. D-Jafari, V. K. Walkerand, and J. D. Palmer. Seven newly discovered intron positions in the triose-phosphate isomerase gene: evidence for the introns-late theory. *Proceedings of the National Academy of Sciences of the United States of America*, 92 (18):8507–8511, August 1995.

A. Lomsadze, V. Ter-Hovhannisyan, Y. O. Chernoff, and M. Borodovsky. Gene identification in novel eukaryotic genomes by self-training algorithm. *Nucleic Acids Research*, 33(20):6494–6506, November 2005.

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

P. Meinicke, M. Tech, B. Morgenstern, and R. Merkl. Oligo kernels for datamining on biological sequences: a case study on prokaryotic translation initiation sites. *BMC Bioinformatics*, 5(1): 169–169, October 2004.

B. Morgenstern. DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics*, 15:211–218, 1999.

M. Pertea, X. Lin, and S. L. Salzberg. Genesplicer: a new computational method for splice site prediction. *Nucleic Acids Research*, 29(5):1185–1190, March 2001.

P. Pollastro and S. Rampone. HS3D, a dataset of homo sapiens splice regions, and its extraction procedure from a major public database. *International Journal of Modern Physics C*, 13(8):1105–1117, 2002.

P. Pollastro and S. Rampone. HS3D: Homo sapiens splice site data set. *Nucleic Acids Research, Annual Database Issue*, 2003.

D. Pyle. *Data preparation for data mining.* Morgan Kaufmann, 2001.

J. C. Rajapakse and L. S. Ho. Markov encoding for detecting signals in genomic sequences. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(2):131–142, 2005.

G. Rautmann and R. Breathnach. A role for branchpoints in splicing in vivo. *Nature*, 315:430–432, May 30-Jun 5 1985.

O. Rinner and B. Morgenstern. AGenDA: Gene prediction by comparative sequence analysis. *In Silico Biology*, 2(3):195–205, 2002.

S. Rogic, A. K. Mackworth, and F. B. Ouellette. Evaluation of gene-finding programs on mammalian sequences. *Genome Research*, 11(5):817–832, May 2001.

S. W. Roy. Recent evidence for the exon theory of genes. *Genetica*, 118(2-3):251–266, July 2003.

Y. Saeys, S. Degroeve, D. Aeyels, P. Rouzé, and Y. Van de Peer. Feature selection for splice site prediction: a new method using EDA-based feature ranking. *BMC Bioinformatics*, 5(1):64, May 2004.

A. A. Salamov and V. V. Solovyev. Ab initio gene finding in Drosophila genomic DNA. *Genome Research*, 10(4):516–522, April 2000.

S. Salzberg, A. L. Delcher, K. H. Fasman, and J. Henderson. A decision tree system for finding genes in DNA. *Journal of Computational Biology : a journal of computational molecular cell biology*, 5 (4):667–680, 1998.

S. L. Salzberg. A method for identifying splice sites and translational start sites in eukaryotic mRNA. *Computer Applications in the Biosciences*, 13(4):365–376, August 1997.

A. F. A. Smit, R. Hubley, and P. Green. Repeatmasker open-3.0. URL `http://www.repeatmasker.org`. (unpublished), 1996-2004.

E. E. Snyder and G. D. Stormo. Identification of coding regions in genomic DNA sequences: an application of dynamic programming and neural networks. *Nucleic Acids Research*, 21(3):607–613, February 1993.

E. E. Snyder and G. D. Stormo. Identification of protein coding regions in genomic DNA. *Journal of Molecular Biology*, 248(1):1–18, April 1995.

V. V. Solovyev, A. A. Salamov, and C. B. Lawrence. Identification of human gene structure using linear discriminant functions and dynamic programming. *Proceedings International Conference on Intelligent Systems for Molecular Biology, ISMB, International Conference on Intelligent Systems for Molecular Biology*, 3:367–375, 1995.

S. Sonnenburg. New methods for splice site recognition, diploma thesis. Master's thesis, Humbold-Universitt zu Berlin, 2002.

R. Staden. Computer methods to locate signals in nucleic acid sequences. *Nucleic Acids Research*, 12 (1 Pt 2):505–519, January 1984.

R. Staden. Methods to define and locate patterns of motifs in sequences. *Computer Applications in the Biosciences*, 4(1):53–60, March 1988.

M. Stanke and B. Morgenstern. AUGUSTUS: a web server for gene prediction in eukaryotes that allows user-defined constraints. *Nucleic Acids Research, Web Server issue*, 33:465–467, July 2005.

M. Stanke and S. Waack. Gene prediction with a hidden markov model and a new intron submodel. *Bioinformatics*, 19 Suppl 2:215–215, October 2003.

M. Stanke, R. Steinkamp, S. Waack, and B. Morgenstern. AUGUSTUS: a web server for gene finding in eukaryotes. *Nucleic Acids Research, Web Server issue*, 32:309–312, July 2004.

A. Stoltzfus, D. F. Spencer, M. Zuker, J. M. Logsdon, and W. F. Doolittle. Testing the exon theory of genes: the evidence from protein structure. *Science*, 265(5169):202–207, July 1994.

G. D. Stormo. Gene-finding approaches for eukaryotes. *Genome Research*, 10(4):394–397, April 2000.

T. Strachan and A. P. Read. *Human Molecular Genetics*. John Wiley & Sons Inc., Second edition, 1999.

J. A. Swets. Measuring the accuracy of diagnostic systems. *Science*, 240:1285–1293, 1988.

L. Taher, O. Rinner, S. Garg, A. Sczyrba, M. Brudno, S. Batzoglou, and B. Morgenstern. AGenDA: homology-based gene prediction. *Bioinformatics*, 19(12):1575–1577, 2003.

L. Taher, B. Morgenstern, and P. Meinicke. Splice site prediction using SVM with the oligo kernel (poster). *ISMB/ECCB*, July 2004a.

L. Taher, O. Rinner, S. Garg, A. Sczyrba, and B. Morgenstern. AGenDA: gene prediction by cross-species sequence comparison. *Nucleic Acids Research, Web Server issue*, 32:305–308, July 2004b.

I. A. Turner, C. M. Norman, M. J. Churcher, and A. J. Newman. Roles of the U5 snRNP in spliceosome dynamics and catalysis. *Biochemical Society Transactions*, 32(Pt 6):928–931, December 2004.

M. G. Tyshenko and V. K. Walker. Towards a reconciliation of the introns early or late views: triosephosphate isomerase genes from insects. *Biochimica et Biophysica Acta*, 1353(2):131–136, August 1997.

E. C. Uberbacher, Y. Xu, and R. J. Mural. Discovering and understanding genes in human DNA sequence using grail. *Methods in Enzymology*, 266:259–281, 1996.

I. Vandenbroucke, T. Callens, A. De Paepe, and L. Messiaen. Complex splicing pattern generates great diversity in human NF1 transcripts. *BMC Genomics*, 3(13):846–857, 2002.

V. Vapnik and A. Chervonenkis. A note on one class of perceptrons. *Automation and Remote Control*, 25, 1964.

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

Vladimir N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1995.

Z. Wang, Y. Chen, and Y. Li. A brief review of computational gene prediction methods. *Genomics, Proteomics & Bioinformatics / Beijing Genomics Institute*, 2(4):216–221, Nov 2004.

J. Weston, A. Elisseeff, B. Schlkopf, and M. Tipping. Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3(7-8):1439–1461, 2003. The data, as well as an extended version of the paper, can be obtained from http://www.kyb.tuebingen.mpg.de/bs/people/weston/l0/.

F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, 1:80–93, 1945.

R. F. Yeh, L. P. Lim, and C. B. Burge. Computational inference of homologous gene structures in the human genome. *Genome Research*, 11(5):803–816, May 2001.

P. D. Zamore, J. G. Patton, and M. R. Green. Cloning and domain structure of the mammalian splicing factor U2AF. *Nature*, 355(6361):609–614, February 1992.

M. Q. Zhang. Identification of protein coding regions in the human genome by quadratic discriminant analysis. *Proceedings of the National Academy of Sciences of the United States of America*, 94(2): 565–568, January 1997.

M. Q. Zhang. Computational prediction of eukaryotic protein-coding genes. *Nature reviews. Genetics.*, 3:698–709, 2002.

M. Q. Zhang and T. G. Marr. A weight array method for splicing signal analysis. *Computer Applications in the Biosciences*, 9(5):499–509, October 1993.

X. H-F. Zhang, K. A. Heller, I. Hefter, C. S. Leslie, and L. A. Chasin. Sequence information for the splicing of human pre-mRNA identified by support vector machine classification. *Genome Research*, 13(12):2637–2650, December 2003.