BIOINFORMATICS ANALYSES OF ALTERNATIVE SPLICING, EST-BASED AND MACHINE LEARNING-BASED PREDICTION

by

JING XIA

B.E., Shanghai Jiaotong Univeristy, China, 2006

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2008

Approved by:

Major Professor
William Hsu

Copyright

Jing Xia

2008

Abstract

Alternative splicing is a mechanism for generating different gene transcripts (called isoforms) from the same genomic sequence. Finding alternative splicing events experimentally is both expensive and time consuming. Computational methods in general, and EST analysis and machine learning algorithms in particular, can be used to complement experimental methods in the process of identifying alternative splicing events. In this thesis, I first identify alternative splicing exons by analyzing EST-genome alignment. Next, I explore the predictive power of a rich set of features that have been experimentally shown to affect alternative splicing. I use these features to build support vector machine (SVM) classifiers for distinguishing between alternatively spliced exons and constitutive exons. My results show that simple, linear SVM classifiers built from a rich set of features give results comparable to those of more sophisticated SVM classifiers that use more basic sequence features. Finally, I use feature selection methods to identify computationally the most informative features for the prediction problem considered.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank Dr. Doina Caragea as my committee member for continuous support. Also, I would like to thank Dr. William H. Hsu as my major professor for guiding my thesis workwith his helpful advice. I also want to be grateful to Dr. Susan Brown at KSU for her guide in my thesis work. This work will not be done without their support. Furthermore, I want to thank Hee Shin Kim at the Bioinformatics Center at Kansas State University, Dr. Caragea and Wesam Elshamy for proof reading the draft and for helpful discussions. I acknowledge the support from the KDD group at Kansas State University. Part of this work was supported by the National Science Foundation under Grant No. 0711396. Last, but not least, I want to thank my little girlfriend Wen for being the greatest.

Dedication

To my parents!

Preface

Hereby, I would like to record my research experience during my master graduate study. The work described in this thesis started as a class project for *the Introduction to Bioinformatics* course that I took at Kansas State University in Spring, 2007. I clearly remember Dr. Caragea (at KSU) mentioning in her class that there are still many open problems in current bioinformatics research and therefore great need for dedicated researchers. Being fascinated by biological pheonomena, I chose one of these topics, pre-mRNA Alternative Splicing, as my class project and continued to work on same problem during my master studies.

Pre-mRNA splicing is a complicated process through which genes remove specific pieces from their sequence, so-called "non-use" RNA, in order to form the final coding messenger RNA. Then, the mRNA is transported to cytoplasm for protein formation. The whole procedure is highly regulated by signals and regulators, known or unknown to scientists, in a single cell. The splicing process can vary a lot under different conditions, resulting in different final products (i.e. which pieces of RNA will be removed can be dependant on different environments).

It is fundamental because around 50% genes of human beings undergo alternative splicing, but there is still no theory for predicting and interpreting the process. What is known is that a number of aspects of the sequence are correlated to AS. For an example, features such as exon and flanking intron size, splice site strength, mRNA secondary structure, or RNA editing, etc, can affect the AS products. With respect to computational approaches, there are exisitng algorithms in machine learning used to guide the computers to learn to predict alternative splicing events. Getting inspiration from two aspects, we tried to incorporate these biological background knowledge into computational approaches, in hope of getting a better classifier. And the results are published as a regular paper in the proceedings of BIBM 2008[XCB08].

Chapter 1

Introduction

In this chapter, we will provide the biological background underlying the process of alternative splicing and then we will present the problems addressed in this work.

1.1 Overview

As well-known, DNA is the genetic code for human heritage. As many species' genomes have been sequenced, the next question is how to find useful knowledge which explains biological phenomenon from these treasure. Central dogma of molecular biology is a process in which gene are transformed from DNA sequence to functional protein products with regulation rules and mechanism involved. One of the steps in the gene expression process is pre-mRNA splicing which refers to the process through which "pieces" of the DNA sequence are removed or retained However, the splicing process is variable, depending on the type of tissue, where it takes place, conditions and so on. Under certain situations, pre-mRNa will remove different pieces from its sequence and splicing together alternatively. Thus, the same gene can result in several different mRNA isoforms products, which are translated into different gene products. Such biological phenomenon is defined as, *Alternative Splicing* (AS). Furthermore, some genes undergo AS events, while others do not. Understanding how to distinguish constitutively spliced genes [1]; from alternatively spliced genes is an important open research topic that scientists are interested in. This involves understanding the effect

[1] constitutively spliced genes are those genes that do not undergo AS events

of the environment in addition to understanding the internal mechanism of AS. The process of AS is one of significant problems which have not been fully understood. Here come the questions: how can we distinguish a AS gene from a constitutively spliced gene how to formalize external environment and internal mechanism in gene splicing and so on.

Thus, identifying AS events using wet lab experiments is both time consuming and expensive, computational approaches have been often used to facilitate the research of AS problems in a different aspect. Here, we apply computational methods, in particular, Support Vector Machines (SVM), to address an important AS problem, specifically, the problem of distinguishing AS genes from constitutively spliced genes. The main idea in this work is to incorporate biological background knowledge of alternative splicing into computational approaches by using an appropriate biological feature representation. Further, we train an inducer to learn to classify genes into two classes, i.e. constitutively and non-constitutively spliced exons. More specifically, here we only consider constitutively and non-constitutively spliced genes in terms of skipped exons, one of AS patterns, i.e. we only consider whether genes contain exons can be skipped during splicing process.

1.2 Biological Background

The modern science of genetics traces its roots to Mendel, who started a new branch in biology, *Genetics*. In 1944, Oswald Theodore Avery, etc. identified the molecule responsible for the transformation of genetics by *deoxyribonucleic acid (DNA)*[AMM44]. James D. Watson and Francis Crick determined the structure of DNA in 1953. One important development was chain-termination DNA sequencing in 1977 by Frederick Sanger: this technology allows scientists to read the nucleotide sequence of a DNA molecule. Recently, the completion of sequencing the genome of the nematode *Caenorhabditis elegans* at the end of 1998 and the first draft sequence of the human genome in June 2000 are significant milestone in history. Table 1.1 is a list of important events in history of genetics.

Hereby, we give a brief introduction to genetics in the following paragraph. There ex-

1865	Genes are particulate factors
1903	Chromosomes are hereditary units
1910	Genes lie on chromosomes
1913	Chromosomes contain linear arrays of genes
1927	Mutations are physical changes in genes
1931	Recombination is caused by crossing over
1944	DNA is genetic material
1945	A gene codes for a protein
1953	DNA is double helix
1958	DNA replicates semiconservatively
1961	Genetic code is triplet
1977	DNA can be sequenced
1997	Genomes can be sequenced
1998	Completion of the genome of the Caenorhabditis Elegans
2000	'Working Draft' of the human genome announced

Table 1.1: *A brief history of genetics*[SRJM02]

ist two kind of organisms in nature, eukaryotes and prokaryotes. The difference between eukaryotic and prokaryotic organisms is that the cells of eukaryotes contain nuclei and a cytoskeleton, while these are not present in prokaryotes. Eukaryotes are considered to be higher organisms since their biological mechanism is much more complicated than those of prokayotes. Human as well as *C. elegans* belong to the class of eukaryotes. Further, in nuclei, genetic information is conserved in form of chromosomes. The whole set of chromosomes, referred as *genome*, is organized as a collection of discrete, separable information packets, called *genes*. That means some pieces of chromosomes are non-informative or have unknown function[Pea06]. For instance, *C. elegans* has six pairs of chromosomes, one pair of which determines the sex. Hermaphrodite *C. elegans*, i.e. which has both male and female reproductive organs, has a matched pair of sex chromosomes (XX); the rare males have only one sex chromosome (X0). More detailed, a series of genes on chromosome X, not the whole chromosome, is correlated to sex determination[Bre74].

Each chromosome is made of a long deoxyribonucleic acids (DNA) chain, which is the carrier of genes and regulatory elements. The DNA chain is constructed by millions of pairwise DNA bases, connecting together pair by pair to form a long double-stranded sequence. The backbone structure of connecting DNAs is made of phosphate and sugar residues. The sugar in DNA is a 2-deoxyribose and pentose (five-carbon) sugar. The sugars are joined together by phosphate groups that form phosphodiester bonds between the third and fifth carbon atoms of adjacent sugar rings. These asymmetric bonds give direction to a strand of DNAs has a direction. In addition to sugar and phosphate, there exist four kinds of nucleotide bases Adenine, Thymine, Guanine and Cytosine in DNA. In Fig. 1.1, there shows a chemical structure of DNA helix. As can be seen in the Figure, only the base pair Adenine-Thymine and Guanine-Cytosine may form by hydrogen bonds. And these pairs, connecting by phosphodiester bonds, form a double helix in 3D structure. The 5' position of one pentose ring to the 3' position of the next 3' pentose ring gives a direction, which is written in 5' to 3' direction. And the two ends of the strand refers to 5' ends and 3' ends with respect to the direction of the strand. Due to the complementary base pairing, the two strands are anti-parallel, cf. Fig. 1.1.

A DNA sequence is a long chain of base pairs (i.e. A-T, C-G) which wrap up and compress, and eventually form a chromosome. A gene can be defined as a piece of region DNA sequence which controls a certain hereditary characteristic. However, most regions on the DNA sequence are non-genetic regions while the current estimated that genes in the human genome is just under 3 billion base pairs and about 20,000 $\sim$ 25,000 protein-coding genes. However, the gene density on the human genome is roughly 12$\sim$15 genes per one megabase pairs. Having explained the DNA structure, next we will briefly introduce the following concepts:

1. the main steps of the gene expression process in eukaryotes

2. biological phenomena of alternative splicing which can be seen as the exception to the standard splicing steps in genes expression process.

At high-level, the process of gene expression in eukaryotes consists the following main steps, Activation of proximal promoters, initiation of transcription, *transcription* and ter-

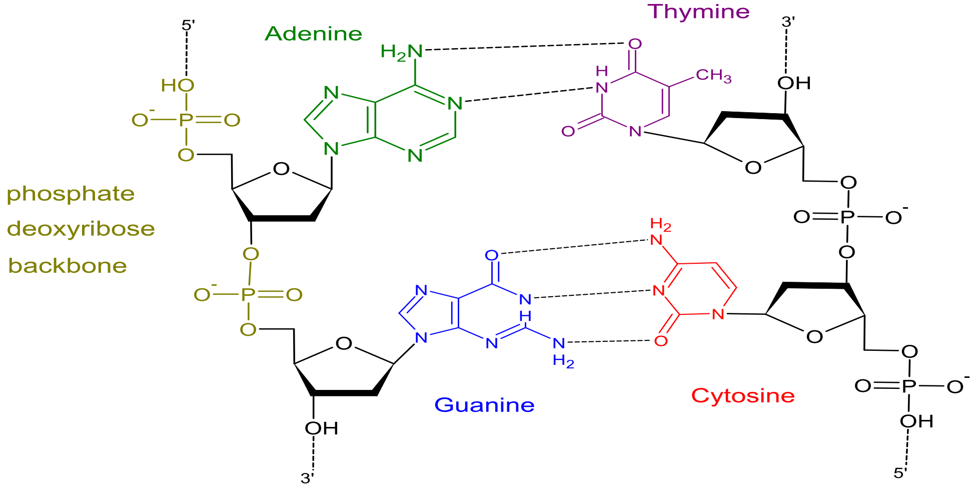


Figure 1.1: *The chemical structure of DNA helix. Four basic DNA molecules, Adenine, Thymine, Guanine, Cytosine, form two pairs (A-T, C-G) which connect through hydrogen bonds,. The two sugar rings on one-stranded side form the backbone which is held together through phosphate bonds. Each strand has a direction. The left-sided strand is from 5' to 3' while the right strand is from 3' to 5' and they are anti-parallel.*

mination, *splicing*, translocation, and *translation*. Table 1.2 comments the steps of the gene expression process and the products in their corresponding phase. These steps will be described into more details in what follows.

Unlike DNA replication, transcription does not need a primer to start. RNA polymerase binds to the DNA with other cofactors, and unwinds the DNA to create a bubble that makes it possible for RNA polymerase to read one of the double-stranded DNA template. In eukaryotes, a collection of proteins called transcription factors mediate the binding of RNA polymerase and the initiation of transcription. Only after certain transcription factors are attached to the promoter, RNA polymerase can also bind. With the proceeding of the RNA polymerase bubble, staring from the Transcriptional Starting Site (TSS) to termination point, a copy of DNA, called precursor messenger ribonucleic acid (pre-mRNA)

Stages	**Products**
1. Activation of proximal (core) promoters and initiation of transcription	Initialized DNA
2. Processing of transcription and termination	pre-mRNA
3. Post-transcriptional modification	mRNA in nucleus
4. Translocation of mRNA to cytoplasm	mRNA in cytoplasm
5. Translation of mRNA into proteins	protein

Table 1.2: *Basic steps of gene expression*

is synthesized; However, the base Thymine in the DNA template is replaced by the base Uracil (U) in the pre-mRNA synthesis.

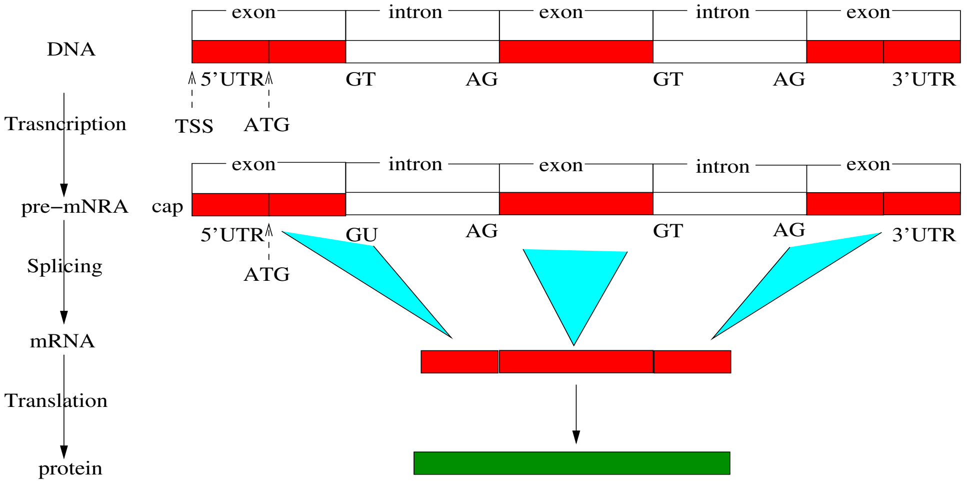


Figure 1.2: *Illustration of main stages of gene expressions*

The next step in the process is the post-transcriptional stage in which the pre-mRNA is transformed into mature mRNA through the *Splicing* mechanism. The pre-mRNA contains non-coding regions called *introns* which are interspersed among coding regions called *exons*. The pre-mRNA sequence starts with an exon and followed by an intron, and then another exon, etc. and it ends with the last exon. Fig. 1.2 shows the main stages of the gene expression process. As we can see, the boundaries between exons and introns, called *splice sites*, have strong consensus sequences. The exon-intron splice sites (i.e., 5' splice sites of introns), called *donor sites*, show a consensus sequence, **GU**, while the other end of introns (i.e., 5' splice sites of introns), called *acceptor sites*, have the sequence of consensus of **AG**. The major process of splicing is a series of protein complexes, called *spliceosomes* as catalysts. With the help of spliceosomes, an intron is bent and move to 3' splice site, forming a loop structure. Next, the 5' splice site is cleaved and bound to one specific site in the intron, called *branch site*, resulting in the formation of an intron lariat structure. The branch site shows a consensus base, **A** and is nearly upstream the 3' splice site. Further, the 3' splice site is further cleaved and the two exons are ligated with the help of spliceosomes. At last, the spliced intron is released and the mRNA is synthesized. Fig. 1.3 illustrates briefly how an intron is spliced out from the sequence.

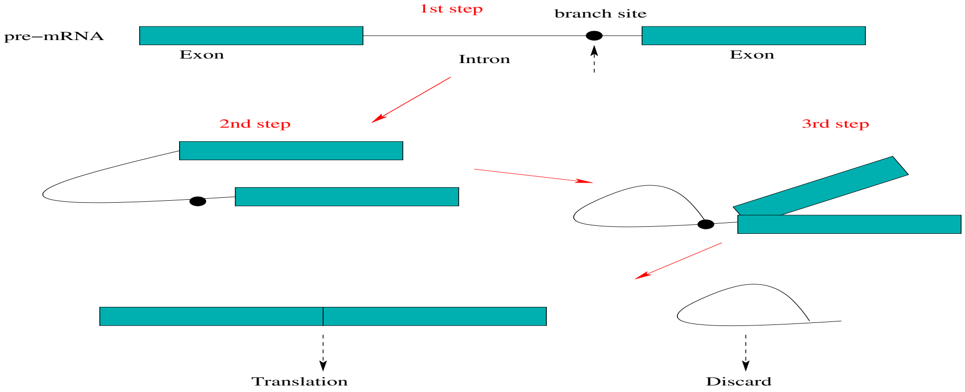


Figure 1.3: *Illustration of the splicing process*

After the pre-mRNA is transformed into mRNA, it will be transported from nucleus to cytoplasm where ribosomes are located. Ribosomes are chemical complexes which surround the mRNAs and function as catalysts in the process of translating the mRNA template into the final products, *proteins*. The mRNA is read in a sequence of triplets, called *codons*. While reading the codon codes, the ribosome and transfer RNA (tRNA) translate the codons into a chain of amino acids until the termination codon is reached. The amino acids join together to form protein, which is a long polypeptide, held together by peptide bonds. Sixty one out of a total sixty four ($4^3 = 64$) possible codon code for twenty amino acids, while the other three combination of bases (**UAG**, **UAA**, **UGA**) are termination codon codes. As the process of DNA transcription, translation also starts at the start codon **AUG**, called *translation initiation site* (TIS). We end here our brief description of the gene expression process and will focus an exception to this process next.

As many precesses, the process of gene expression has a lot of exceptions in nature, and *pre-mRNA alternative splicing* is among one of these well-known exceptions. For years, it was believed that one gene corresponds to one protein, but the discovery of alternative splicing[Gil78] provided a mechanism for generating different gene transcripts (isoforms) from the same genomic sequence. Fig. 1.5 illustrates five main kinds of AS patterns and shows how isoforms differentiate from each other. Years after its discovery, alternative splicing was still seen more as the exception than the rule[Ast04]. Recently, however, it has become obvious that a large fraction of genes undergoes alternative splicing[Gra01]. Early analyses suggested that at least 50% of human genes undergo alternative splicing[int,ven]. More recently, approximately 75% of human genes appear to be alternatively spliced[JCGE+03]. Alternate splicing occurs in many other organisms. Approximately 20% of the predicted genes in Drosophila species have been shown to undergo alternative splicing[LTR04], some of which can produce a very large number of isoforms. For example, Drosophila's Dscam gene could in principle produce up to 38,016 isoforms[SCS+00] and this diversity is essential for neuronal wiring and self-recognition[DEH+07]. Another example is the *unc-32O* gene in model organism, *C. elegans* that can be alternatively spliced to six different transcripts isoforms (1.4). As shown in the figure, exon 4 can undergo three different mutually exclusive AS, while two for exon 7 leading to six possible transcripts[Zah05]. On the other hand, aberrant splicing has been linked to pathological states such as cancer[KTB+02]. These results underscore the importance of alternative splicing, both in normal and aberrant conditions. However, the task of accurately identifying alternative splicing isoforms is particularly intricate, as different transcriptional isoforms can be found in different tissues or cell types, at different development stages or induced by external stimuli.

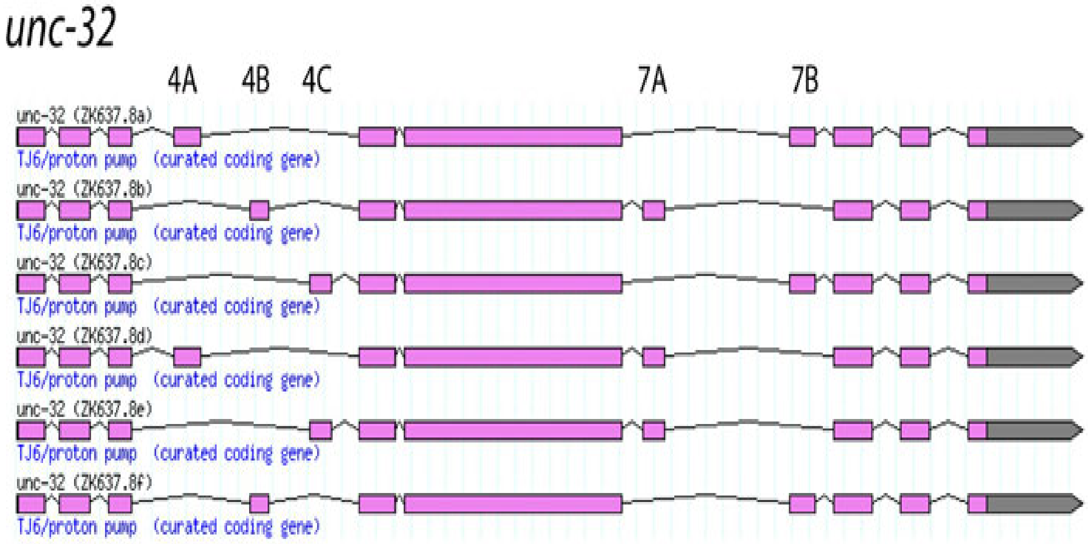

unc-32
4A
4B
4C
7A
7B
unc-32 (ZK637.8a)
TJ6/proton pump (curated coding gene)
unc-32 (ZK637.8b)
TJ6/proton pump (curated coding gene)
unc-32 (ZK637.8c)
TJ6/proton pump (curated coding gene)
unc-32 (ZK637.8d)
TJ6/proton pump (curated coding gene)
unc-32 (ZK637.8e)
TJ6/proton pump (curated coding gene)
unc-32 (ZK637.8f)
TJ6/proton pump (curated coding gene)

Figure 1.4: *Example of alternatively spliced genes in C. elegans. Illustration of unc-32 in C. elegans undergo six alternative splicing. Images are taken from Wormbase genome browser*

1.3 Problem specification

The task of accurately identifying alternative splicing isoforms is particularly intricate, as different transcriptional isoforms can be found in different tissues or cell types, at different development stages, or can be induced by external stimuli. Experimental methods for finding alternative splicing events are expensive and time consuming. Therefore, computational methods that can complement experimental methods are needed. Traditional computational methods rely on aligning expressed sequence tags (ESTs) and complementary DNA (cDNA) to genomic DNA to identify alternative splicing events[NGR06,KRGS01]. The basic idea of the approach is based on the alignment transcripts to genome sequences, which identifies the gene loci and structures. AS events can be identified through further scanning the boundaries of the genes. However, this approach is limited to both the quality and the coverage of the transcripts. More recently, machine learning approaches is used to predict alternative splicing events through "learning" various sequence features[RSS05a,WM06,SSC+04].

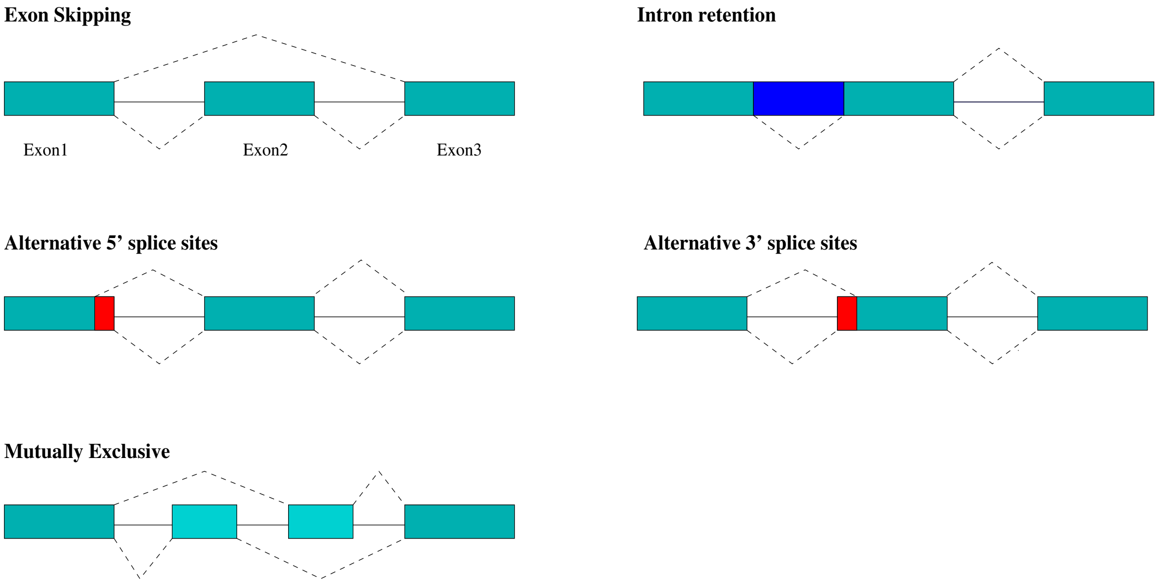


Figure 1.5: *Illustration of five kinds of alternative splicing patterns. Lower-level dash lines connect exons, forming constitutively spliced isoforms, while upper-level dash lines connect exons alternatively, making an alternative spliced isoforms. According to its different spliced structure, five AS patterns are Exon Skipping, Intron Retention, Alternative 3' Splice Sites, Alternative 5' Splice Sites and Exon Mutually Exclusive*

Although several types of alternative splicing events exist (e.g., alternative acceptor, alternative donor, intron retention), in this thesis we focus on the prediction of cassette exons, one particular type of splicing event, where an exon is a cassette exon (or alternatively spliced) if it appears in some mRNA transcripts, but does not appear in all isoforms. If an exon appears in all isoforms, then it is called a constitutive exon. Several basic sequence features have been used to predict if an exon is alternatively spliced or constitutive, including: exon and flanking introns lengths and the frame of the stop codon. In particular, G. Rätsch et al.[RSS05a] have proposed a kernel method, which takes as input a set of local sequences represented using such basic features and builds a classifier that can differentiate between alternatively spliced and constitutive exons. In the process of building the classifier, this method identifies and outputs predictive *splicing motifs*, which are used to interpret the results. In this context, a motif is a sequence pattern that occurs repeatedly in a group of related sequences. The method in the work[RSS05a] is essentially searching for motifs within a certain range around each base. This range needs to be carefully chosen in order to obtain good prediction results[HO08].

Finding motifs that explain alternative splicing of pre-mRNA is not surprising as it has been experimentally shown that alternative splicing is highly regulated by the interaction of intronic or exonic RNA sequences (more precisely, motifs that work as signals) with a series of splicing regulatory proteins[HO08]. Such splicing motifs can provide useful information for predicting alternative splicing events, in general, and cassette exons, in particular. Generally, computational identification of splicing motifs can be derived from patterns that are conserved in another organism[KBSM+06,SA03,DSS05b]. However, since some exons and most introns are not conserved, it is desirable to identify such motifs directly from local sequences in the organism of interest.

In addition to motifs, several other sequence features have been shown to be informative for alternative splicing prediction[HO08]. Among these, pre-mRNA secondary structure has been investigated to identify patterns that can affect splicing[HZBS07,PYR02]. It has been found that the pre-mRNA exhibits local structures that enhance or inhibit the hybridization of spliceosomal snRNAs to the pre-mRNA. In other words, the structure can affect the selection of the splice sites. As another feature, the strength of the general splice sites is very important with respect to the splicing process, as strong splice sites allow the spliceosomes to recognize pairs of splice sites between long introns[WM06,FH07]. When the splice sites degenerate and weaken, other splicing regulatory elements (exon/intron splicing enhancers and silencers)[PMS07] are needed. At last, one other feature that has been shown to be correlated with the spicing process is given by the base content in the vicinity of splice sites[HO08].

Although the method in the work[RSS05a] can *output* motifs that explain the classifier results, to the best of our knowledge there is no study that explores motifs (derived either using comparative genomics or local sequences) and other alternative splicing features (pre-mRNA secondary structure, splice site strength, splicing enhancers/silencers and base content) together as *inputs* to machine learning classifiers for predicting cassette exons.

In this thesis, we use the above mentioned features with state-of-the-art machine learning methods, specifically the SVM algorithm, to generate classifiers that can distinguish alternatively spliced exons from constitutive exons. We show that the classification results obtained using all these features with simple linear SVMs are comparable and sometimes better than those obtained using only basic features with more complex non-linear SVMs. To identify the most discriminative features among all features in our study, we use machine learning methods (SVM feature importance and information gain) to perform feature selection.

1.4 Organization

The rest of the thesis is organized as follows: We have introduced biological background relevant to our issue in Chapter 1.2, we will introduce computational methods to identify alternative splicing events, the machine learning algorithms used to predict alternatively spliced exons and to perform feature selection. In Chapter 3, we describe the data set used in our experiments and explain how we construct the features considered in our study. We present experimental results in Chapter 4 and conclude with a summary and ideas for future work in Chapter 5.

Chapter 2

Methods

2.1 Identifying AS events by analyzing EST data

Expressed sequence tags (EST) are short, around 200 $\sim$ 800 nucleotide bases in length, unedited, randomly selected single-pass sequence reads derived from cDNA library[NGR06]. Due to both the cost-effectiveness of generating ESTs and the value of the biological information, high-throughout of ESTs are used in a wide area of biology (gene discovery, gene structure identification, alternative transcripts detection, complement of genome annotation, single nucleotide polymorphism characterization and facilitation of proteome analysis)[NGR06]. On the other hand, the error-prone property of ESTs (caused by single pass and errors toward the ends of the reads) might cause false prediction of knowledge.

When ESTs are used to investigate alternative splicing, *EST-mRNA comparisons*, *EST-genome alignment comparisons*, and *EST-genome multiple alignment comparisons* are three computational methods and the latter two being more robust with respect to prediction accuracy[BRP06]. Overall, the main idea underlying all three kinds of methods is 1) to identify the gene structure by alignments of ESTs with genomic or transcript sequences, and then 2) to predict alternative splicing by looking for different transcript isoforms (i.e. patterns of alternative splicing) of one gene structure. A number of tools are available for automating the procedure of alignment of ESTs to genomic DNA (listed in Tabel 2.1). The key algorithm underlying the sequence alignment is *dynamic programming*, DP, which can compute the

Name	**Website**	**Category**[a]
ASPIC*	http://t.caspur.it/ASPIC/	W
BLAT*	http://genome.ucsc.edu/FAQ/FAQblat	F,W
DDS/GAP2	http://www.tigr.org/software/alignment.shtml	F,W
EST_GENOME	http://www.ebi.ac.uk/~guy/exonerate/	F
EXONERATE*	http://www.ebi.ac.uk/~guy/exonerate/	F
EXALIN	http://blast.wustl.edu/exalin	F
GENESEQER*	http://deepc2.psi.iastate.edu/cgi-bin/gs.cgi	F,W
GMAP*	http://www.gene.com/share/gmap	F
MGALIGN	http://origin.bic.nus.edu.sg/mgalign/	F,W
MRNAVSGEN	http://genes.mit.edu/genoa	F
SIM4*	http://globin.cse.psu.edu/html/docs/sim4.html	F
SPIDEY	http://www.ncbi.nlm.nih.gov/spidey	F,W
SPLIGN*	http://www.ncbi.nlm.nih.gov/sutils/splign/	F,W
WABA	http://www.soe.ucsc.edu/~kent/xenoAli/	F,W

[a] F, free for academic users; W, web server avaiable; *, investigated in the research

Table 2.1: *Programs for spliced-sequence to genome DNA alignment*

optimal sequence alignment in quadratic runtime and memory complexity. For an example, given an mRNA sequence and a genomic sequence, DP finds an alignment of the mRNA to the genomic sequence, which allows long gaps insertion into the mRNA sequence. Thus, gap penalties should be based on intron length distribution, and gaps following the rules of splice sites should be more preferred[HO08].

After genome-wide alignments are performed, the next step is the identification of constitutive and alternative exons as one type of alternative splicing patterns. Constitutive exons are in default splicing isoforms, while exons that are as annotated alternatively spliced, should meet specific conditions. After identifying the splicing junctions (i.e. putative splicing sites), one can order them with respect to gene loci to construct a matrix of splicing junctions, in which exons are considered as nodes and splicing junctions as edges connecting nodes. Traversing through this adjacent-matrix graph will capture the alternative splicing events[KSG02].

This approach based on ESTs analysis is widely used for identifying AS events. However, there are some limitations to the approach, which are mainly from both the error-prone property of ESTs and the limited coverage of ESTs for newly sequenced genome. These limitations will result in making false positive identification of AS events and missing AS events. And this approach can not be used to predict AS events for genes. Thus, ,ore recently, machine learning based approaches are used to do the task of predicting AS events. Next, we will introduce one of the machine learning algorithms, Support Vector Machines (SVM) and how to adapt this advanced algorithm to the problem of predicting AS events.

2.2 Support Vector Machine Classifiers

The support vector machine (SVM) algorithm[Vap99] is one of the most effective machine learning algorithms for many complex binary classification problems, including a wide range of bioinformatics problems[GWBV02,LEC+03,BHB03,PMS07], and has been recently used to detect splice sites[RS04,RSS05a,SSP+07]. The SVM algorithm takes as input labeled data from two classes and outputs a model (a.k.a., classifier) for classifying new unlabeled data into one of those two classes. SVM can generate linear and non-linear models.

Let $E = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_l, y_l)\}$, where $\mathbf{x}_i \in R^p$ and $y_i \in \{-1, 1\}$, be a set of training examples. Suppose the training data is *linearly separable*. Then it is possible to find a hyperplane that partitions the pattern space into two half-spaces. Suppose there is one for the linearly-separable data set. The classifier is defined as a function $f : \mathbf{x} \rightarrow \{-1, +1\}$ that predicts the label y_i of any $\mathbf{x}_i \in R^p$. SVM algorithm learns to construct above defined decision function from the set of training examples, E in the form of a linear separating hyperplane. We have

$$\mathbf{f}(\mathbf{x}, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b) \tag{2.1}$$

The set of such hyperplanes is given by $\{\mathbf{x} | \mathbf{x} \cdot \mathbf{w} + b = 0\}$, where $\mathbf{x}$ is the p-dimensional data vector and $\mathbf{w}$ is the normal to the separating hyperplane. The learning task is to find the normal weights $\mathbf{w}$ and the bias b.

2.2.1 The Separable Case

Firstly, we assume that the training examples can be separated by a linear hyperplane as defined in Eq. (2.1). SVM selects among the hyperplanes that correctly classify the training set, the one that maximizes the margin between positive and negative examples, subject to certain constraints such that:

$$\mathbf{w}\cdot\mathbf{x}+b \geq +1 \; for \; y_i=+1 \tag{2.2}$$

$$\mathbf{w}\cdot\mathbf{x}+b \leq -1 \; for \; y_i=-1 \tag{2.3}$$

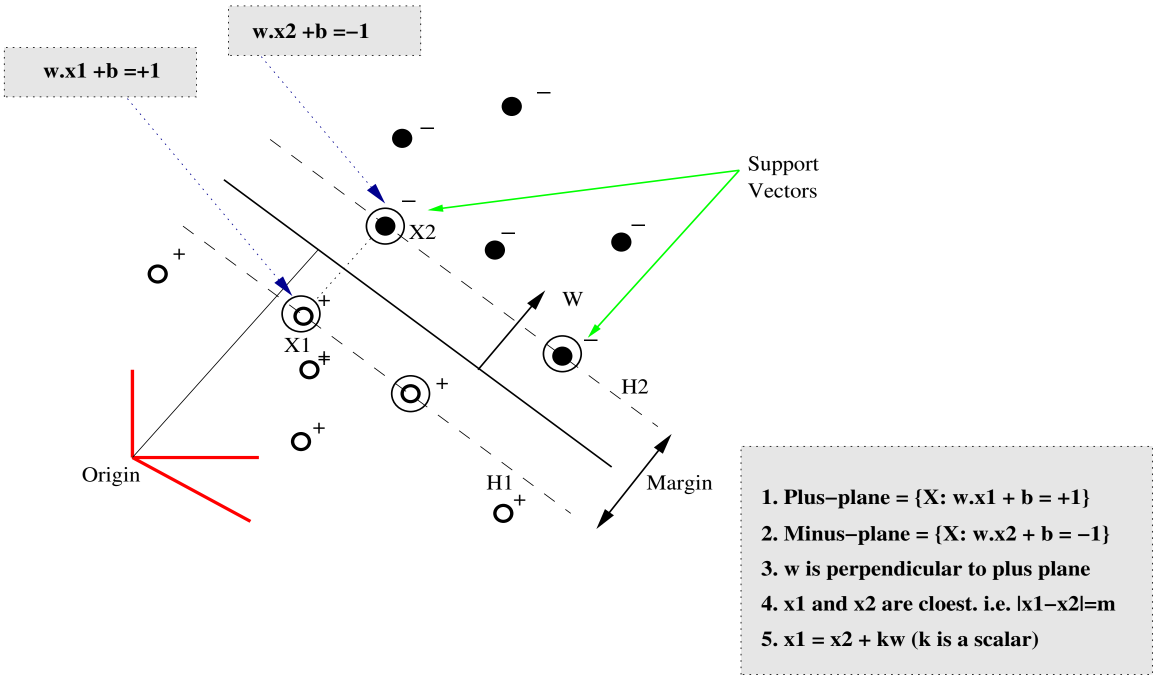
$$\Rightarrow y_i(\mathbf{x_i}\cdot\mathbf{w}+b) \geq \pm 1 \; for \; y_i=\pm 1 \tag{2.4}$$

We denote the plus hyperplane as H_1 and minus hyperplane as H_2, both of which are parallel to the decicsion boundary $H=\mathbf{x}\cdot\mathbf{w}+b=0$.

$$\begin{aligned} H1 &: \; \mathbf{w}\cdot\mathbf{x}+b=+1 \\ H2 &: \; \mathbf{w}\cdot\mathbf{x}+b=-1 \end{aligned}$$

The perpendicular distance between these hyperplanes is given by $\frac{2}{||\mathbf{w}||}$, which is called *margin*. SVM chooses one minimal normal $||\mathbf{w}||$ which maximizes the margin and also satisfies the constraint Eq. 2.4. For an example of support vector machines, Fig 2.1 shows maximal margin between the hyperplanes of two classes. We derive formula $\frac{2}{||\mathbf{w}||}$, the distance of margin, from it[Bur98].

What we know:	**Induction:**	**Induction Cont. :**
(1) $\mathbf{w}\cdot\mathbf{x_1}+b=+1$	Based on (1) & (3)	Based on (4) & (5)
(2) $\mathbf{w}\cdot\mathbf{x_2}+b=-1$	$\Rightarrow \mathbf{w}\cdot(\mathbf{x_2}+\lambda\mathbf{w})+b=1$	$M=\lvert\mathbf{x_1}-\mathbf{x_2}\rvert=\lvert\lambda\mathbf{w}\rvert$
(3) $\mathbf{x_1}=\mathbf{x_2}+\lambda\mathbf{w}$	$\Rightarrow \mathbf{w}\cdot\mathbf{x_2}+b+\lambda\mathbf{w}\cdot\mathbf{w}=1$	$=\lambda\lvert\mathbf{w}\rvert=\lambda\sqrt{\mathbf{w}\cdot\mathbf{w}}$
(4) $\lvert\mathbf{x_1}-\mathbf{x_2}\rvert=M$	$\Rightarrow -1+\lambda\mathbf{w}\cdot\mathbf{w}=1$	$=\frac{2\sqrt{\mathbf{w}\cdot\mathbf{w}}}{\mathbf{w}\cdot\mathbf{w}}=\frac{2}{\sqrt{\mathbf{w}\cdot\mathbf{w}}}$
	$\Rightarrow \lambda=\frac{2}{\mathbf{w}\cdot\mathbf{w}}$ (5)	$=\frac{2}{\lvert\mathbf{w}\rvert}$



1. Plus−plane = {X: w.x1 + b = +1}
2. Minus−plane = {X: w.x2 + b = −1}
3. w is perpendicular to plus plane
4. x1 and x2 are cloest. i.e. |x1−x2|=m
5. x1 = x2 + kw (k is a scalar)

Figure 2.1: *Illustration of support vectors and margin*[Bur98]

Maximum margin reasons 1) Intuitively, the larger the margin is, the safer it is. 2) If we have made a small error in the location of the boundary, maximum margin gives us the least chance of causing a misclassification. 3) Make leave-one-out cross-validation (LOOCV) easy because the model is immune to removal of any non-support-vector data points. 4) Theory using Vapnik-Chervonenkis (VC) dimension supports the proposition that maximum margin is good. 5) Empirically it works very well. 6) For linear classifiers without bias, i.e. $b = 0$, it has been proven in the work[BSt] that the test error has an upper boundary given by a term of the sum of the fraction of training examples within a certain margin ρ and proportional to a term of $\frac{R}{\rho}$ (R denotes the smallest radius of a sphere containing all samples). Therefore, with a fixed R, the test error will be minimized when the SVM chooses a maximum margin for a certain number of training examples.

To maximise the margin $\frac{||\mathbf{w}||^2}{2}$ with respect to the constraints in Eq. (2.4). We need to solve the following convex quadratic optimisation problem:

To solve this problem, a Lagrangian formulation of the problem will be introduced be-

$$\begin{array}{ll} \text{maximize} & \frac{||\mathbf{w}||^2}{2} \\ \text{subject to constraints} & c_i \equiv y_i(\mathbf{x_i} \cdot \mathbf{w} + b) - 1 \geq 0,\ \ i = 1, ..., l \end{array}$$

cause 1) the reformation is easier to handle in terms of mathematics than the formulation with constraints in Eq. (2.4). 2) The training samples will only appear in the form of dot products between vectors and make the same approach feasible when the data is the non-linearly separable.

The algorithm assigns a weight $\alpha_i \geq 0$, $i = 1, ..., l$ called *positive Lagrange multipliers*, to each constraints $c_i,$, $i = 1, ..., l$. Mathematically, each constraint equations, i.e. c_i, are multiplied by α_i and subtracted from the objective function $\frac{||\mathbf{w}||^2}{2}$, to form the Lagrangian formulation.

$$L_P \equiv \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{l} \alpha_i y_i (\mathbf{x_i} \cdot \mathbf{w} + b) + \sum_{i=1}^{l} \alpha_i \tag{2.5}$$

To achieve maximum margin, L_P must be minimized with respect to $\mathbf{w}$, b, which means the derivative of L_P with respect to all α_i to equal zero, subject the constraints $\alpha_i \geq 0$.

Because the above problem is a convex quadratic programming problem, and the objective function is itself convex. And those points which satisfy the constraints also form a convex, proof results that the problem can be equivalently solved as a dual problem: *maximize* L_P, subject to the constraints that the $\frac{\partial L_P}{\partial \mathbf{w}} = 0$ and $\frac{\partial L_P}{\partial b} = 0$, also subject to the constraints that all $\alpha_i \geq 0$. To distinguish it from the previous one, it is denoted as L_D. It has a property called "Wolfe's Dual" that the solution to maximize L_D in the form of $\mathbf{w}$, b, α has the same value as the primary problem which minimize the L_P. Based on the above constraints that the gradient of L_P with respect to $\mathbf{w}$ and b vanish, we get the conditions:

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x_i} \tag{2.6}$$

$$\sum_i \alpha_i y_i = 0 \tag{2.7}$$

Since the property of dual problems, (i.e. equality constraints), Eq. 2.6 and Eq. 2.7 can be substituted into Eq. 2.5 and we get:

$$
\begin{aligned}
L_D &= \frac{1}{2}||\sum_i \alpha_i y_i \mathbf{x_i}||^2 - \sum_{i=1}^{l} \alpha_i y_i (\mathbf{x_i} \cdot \sum_i \alpha_i y_i \mathbf{x_i}) + \sum_{i=1}^{l} \alpha_i \\
&= \frac{1}{2}(\sum_i \alpha_i y_i \mathbf{x_i} \cdot \sum_i \alpha_i y_i \mathbf{x_i}) - \sum_i \alpha_i y_i \mathbf{x_i} \cdot \sum_i \alpha_i y_i \mathbf{x_i} + \sum_i \alpha_i \\
&= \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j}
\end{aligned}
$$

The remaining task is to maximize L_D with respect to the $\alpha_i \geq 0$, subject ot constraints 2.7. As mentioned, each training data has a weight α_i. After support vector training, each weights are assigned a value. The points having non-zero weight are called *support vectors* and lie either on the hyperplane H_1 or H_2. Those points of which weights are non-zero lie on either side of H_1 or the side of H_2.

2.2.2 The Karush-Kuhn-Tucker Conditions

The Karush-Kuhn-Tucker Conditions (KKT) are very importance for any constrained optimization problems (convex or not), with any kind of constraints and thus for the primal problem (i.e. L_P) mentioned above, the KKT conditions are stated as:

$$\frac{\partial}{\partial w_v} L_P = w_v - \sum_i \alpha_i y_i x_i v = 0 \qquad v = 1, ..., d \tag{2.8}$$

$$\frac{\partial}{\partial b} L_P = -\sum_i \alpha_i y_i = 0 \tag{2.9}$$

$$y_i(\mathbf{w} \cdot \mathbf{x_i} + b) - 1 \geq 0 \qquad i = 1, ..., l \tag{2.10}$$

$$\alpha_i \geq 0 \qquad \forall i \tag{2.11}$$

$$\alpha_i(y_i(\mathbf{w} \cdot \mathbf{x_i} + b) - 1) = 0 \qquad \forall i \tag{2.12}$$

As the *necessary and sufficient* conditions for any solutions to SVM, the KKT conditions hold for all support vector machines? and this fact results in a set of methods to find the solution. For example, while **w** needs to be found during the training procedure, the threshold b can be derived from KKT conditions, using "complementarity" condition of Eq. 2.12. By choosing one k for which $\alpha_k \neq 0$, we get

$$\begin{aligned} &\alpha_k(y_k(\mathbf{w}\cdot\mathbf{x_k}+b)-1)=0 \qquad \alpha_k \neq 0 \\ \Rightarrow &y_k(\mathbf{w}\cdot\mathbf{x_k}+b)-1=0 \\ \Rightarrow &b=\frac{1}{y_k}-\mathbf{w}\cdot\mathbf{x_k} \qquad\qquad y_k \neq 0 \end{aligned}$$

The separating hyperplane is defined as a weighted sum of support vectors. Thus, $\mathbf{w} = \sum_{i=1}^{l}(\alpha_i y_i)\mathbf{x_i} = \sum_{i=1}^{s}(\alpha_i y_i)\mathbf{x_i}$, where s is the number of support vectors, y_i is the known class for example $\mathbf{x}_i$, and α_i are the support vector coefficients that maximize the margin of separation between the two classes. The classification for a new unlabeled example can be obtained from

$$\begin{aligned} f_{\mathbf{w},b}(\mathbf{x}) &= \text{sign}(\mathbf{w}\cdot\mathbf{x}+b) = \text{sign}(\mathbf{w}\cdot\mathbf{x}-\mathbf{w}\cdot\mathbf{x_k}+\frac{1}{y_k}) \\ &= \text{sign}(\mathbf{w}\cdot(\mathbf{x}-\mathbf{x_k})+\frac{1}{y_k}) \\ &= \text{sign}\left(\sum_{i=1}^{l}\left(\alpha_i y_i \mathbf{x_i}\cdot(\mathbf{x}-\mathbf{x_k})+\frac{1}{y_k}\right)\right) \end{aligned}$$

2.2.3 The Non-separable Case

If the goal of the classification problem is to find a linear classifier for a non-separable training set (e.g., when data is noisy and the classes overlap), we can relax the constraints 2.2 and 2.3 by introducing a set of *slack variables*, ξ_i, $i = 1, ..., l$ to allow for the possibility of examples violating the constraints $y_i(\mathbf{x_i}\cdot\mathbf{w}+b) \leq 1$. Then we have:

$$\mathbf{w}\cdot\mathbf{x}+b \geq +1-\xi_i \ for \ y_i=+1 \tag{2.13}$$

$$\mathbf{w}\cdot\mathbf{x}+b \leq -1+\xi_i \ for \ y_i=-1 \tag{2.14}$$

$$\xi_i \geq 0 \ for \ \forall i \tag{2.15}$$

In this case the margin is maximized, paying a penalty proportional to the cost C of constraint violation, i.e., $C\sum_{i=1}^{l}\xi_i$, which is an upper bound on the number of training errors.

The objective function to be minimized will be changed from $||w||^2$ to $||w||^2 + C(\sum_i \xi_i)^k$, where C is a parameter to be chosen by the user and a larger C indicates to assign a higher penalty to errors. Here, for any positive integer k, the above optimization problem is a convex programming problem. However, if we choose $k = 1$, it is also a quadratic programming problem and furthermore the Lagrangian formulation in the Wolfe dual problem does not have the terms of ξ_i, which becomes:

Maximize:

$$L_D \equiv \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j} \tag{2.16}$$

subject to:

$$0 \leq \alpha_i \leq C, \;\; for \; \forall i \tag{2.17}$$

$$\sum_i \alpha_i y_i = 0. \tag{2.18}$$

This case is illustrated in Fig. 2.2

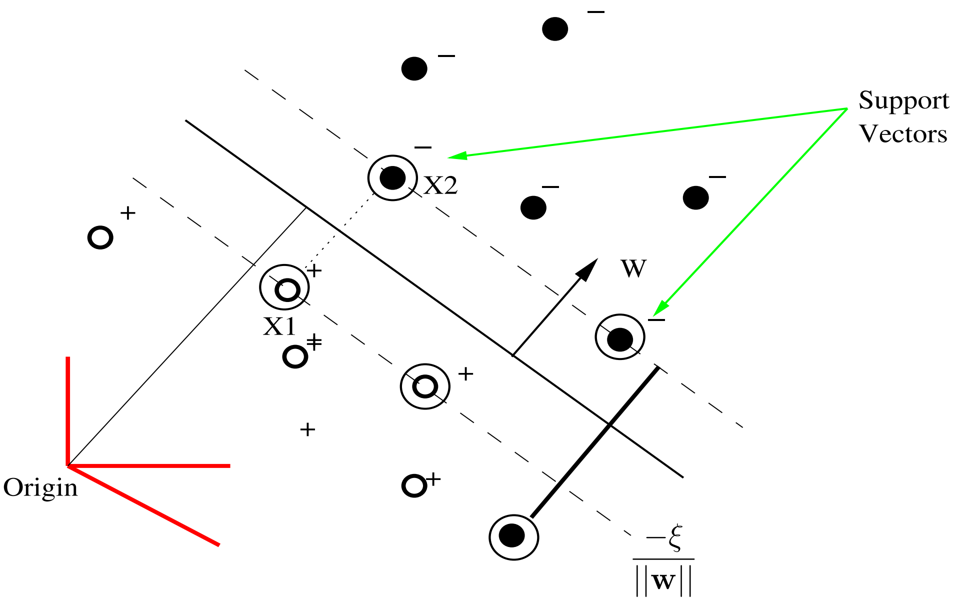


Figure 2.2: *Linear hyperplane for non-linear separable case*

2.2.4 Nonlinear Support Vector Machines

If the decision function is not a linear function of the data, the SVM works by mapping the training set into a higher dimensional *feature* space, $\Phi : \mathbb{R}^D \rightarrow H$, where the data becomes linearly separable, using an appropriate kernel function K.

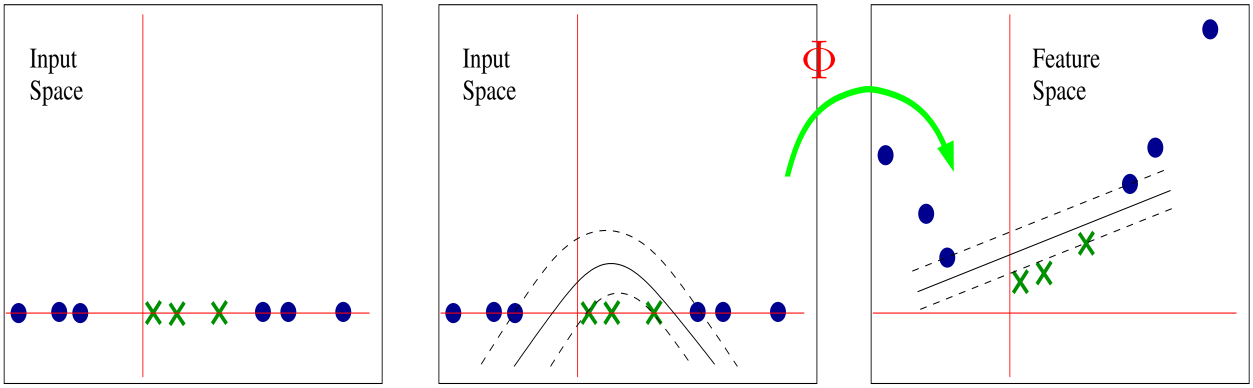


Figure 2.3: *Example of mapping input spaces to feature spaces. The leftmost figure shows that the data set is in one dimension, which can not be separated by a linear hyperplane; the middle one shows that a non-linear hyperlane separates the data set with no errors. When mapped to a two-dimenion space, the data set is separated by a linear hyperlane in rightmost picture.*

Since in the training procedure and in the decision function only dot products $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$ need to be calculated, we only need to know what the dot product is, without need of even knowing what is the mapping function, Φ. Thus, we define K, the so-called "kernel function" such that $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$ and Eq. 2.16 transforms to

$$L_D \equiv \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{2.19}$$

and the decision function changes to

$$f_{\mathbf{w},b}(\mathbf{x}) = \text{sign}(\sum_i \alpha_i y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}) + b) = \text{sign}(\sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b) \tag{2.20}$$

To be valid kernels, it has proven that the kernels should satisfy Mercer's condition[?] , where there exists a mapping Φ and an expansion $K(\mathbf{x},\mathbf{y}) = \sum_i \Phi(\mathbf{x}_i)\Phi(\mathbf{y}_i)$ if and only if, for any $g(\mathbf{x})$ such that $\int g(\mathbf{x})^2 d\mathbf{x}$ is finite, then $\int K(\mathbf{x},\mathbf{y})g(\mathbf{x})g(\mathbf{y})d\mathbf{x}d\mathbf{y} \geq 0$

Kernel Examples

1. *Polynomial kernel*:

$$K(\mathbf{x},\mathbf{y}) = (\mathbf{x}\cdot\mathbf{y}+1)^p \tag{2.21}$$

2. *Gaussian radial basis function (RBF) kernel*:

$$K(\mathbf{x},\mathbf{y}) = e^{-||\mathbf{x}-\mathbf{y})||/2\sigma^2} \tag{2.22}$$

For specific computational problems, the choice of a good kernel function is of importance. For alternative splicing, in previous work[DSS05b] use linear kernel with some basic biological features (we will state more about feature in next chapter) to get a good result. To achieve a more higher accuracy, in the work[RSS05a], a kernel function, called as *shift weighted degree kernel*, preforms well in predicting one specific isoform of alternative splicing. We use the LIBSVM implementation of SVM, available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/, in our study.

2.2.5 Weighted Degree Kernel

The weighted degree kernel is a kind of measure for the similarity of pairwise sequence. The more similar two DNA sequences are, the higher the value of the kernel function is. The main idea is to count the number of occurrences of $k$mers appearing in both sequences $s1$ and $s2$, which share the same length L. The degree, denoted d, is defined as the maximum length of $k$mers matches between two sequences (i.e. $k \in \{1,...,d\}$). The weights capture the idea that longer matches of $k$mers contribute more significance to the kernel value. With respect to above ideas, the initial mathematical formulation introduced in the work[SSP+07], given in 2.23

$$k(\mathbf{s}_i,\mathbf{s}_j) = \sum_{k=1}^{d}\beta_k \sum_{l=1}^{L-k+1} \mathbf{I}(\mathbf{u}_{k,l}(\mathbf{s}_i) = \mathbf{u}_{k,l}(\mathbf{s}_j)) \tag{2.23}$$

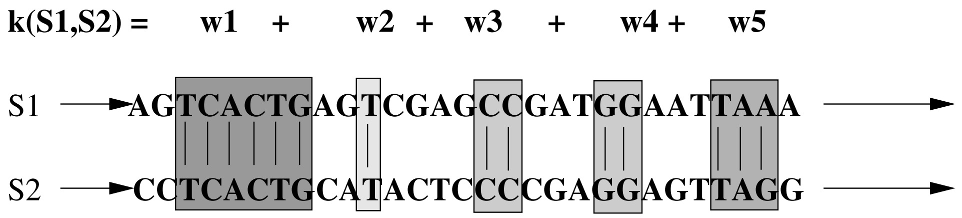

k(S1,S2) = w1 + w2 + w3 + w4 + w5
S1
AGTCACTGAGTCGAGCCGATGGAATTAAA
S2
CCTCACTGCATACTCCCCGAGGAGTTAGG

Figure 2.4: *Given two sequences of same length s1 and s2, the kernel value is the sum of all weights, each of which has a value depending on the length of matches, where longer matches contribute more value than smaller ones.*

Here, the identity function $\mathbf{I}(term)$ is defined as: if term's value equals **true**, then $\mathbf{I} = 1$ otherwise $\mathbf{I} = 0$. ($\mathbf{u}_{k,l}(\mathbf{s})$ is the $k$mer starting from position l of the sequence $\mathbf{s}$). Thus the inner summation corresponds to counting the matching occurrences of $k$mer. And β_k is the weight coefficients assigned to each $k$mer, where mathematically $\beta_k = 2(d-k+1)/(d(d+1))$[SRS05]. A note should be mentioned that β_k is a decreasing function of k (d is constant). This property of β_k does not violate the assumption that longer matches contribute more strongly than smaller ones because long matches implies all summation of β of the shorter matches in 2.23. Furthermore, another trick for speeding up the kernel computation can be derived from the above observation: instead of calculating β_k in 2.23 for each matches, the maximal length of $k$mer can be found first and an overall weight coefficient, so-called reward r_b can be assigned to the maximal block. Figure 2.4 shows an example of how to compute the kernel value based on the two sequences. We give formulation of r_b here.

$$r_b = \begin{cases} (b(3db+3d-b^2+1)/3d(d+1)) & \text{for } b \leq d \\ (b-d)(3-1/3) & \text{for } b > d. \end{cases} \tag{2.24}$$

The WD kernel works well when the positions of two motifs are approximately same on two sequences. However, if one of the two motifs shifts one base in the sequence, the WD kernel will miss the matching. Therefore, in the work[RSS05b] the authors extend WD kernel in order to find such matching motifs with some-base shifts (Fig. 2.5 illustrates the situation in which two motifs shift each other). The *weighted degree kernel with shifts* is defined as

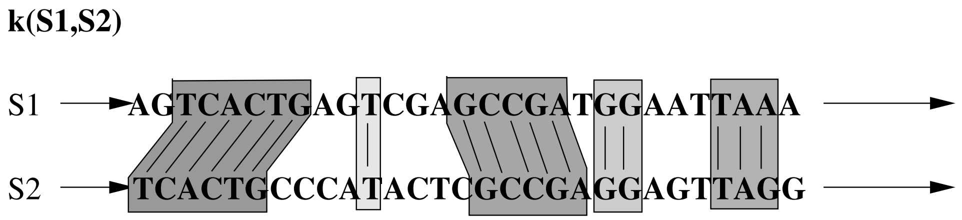


Figure 2.5: *Given two sequences of same length s1 and s2, the motifs have shifted each other in extent of several bases.*

$$k(\mathbf{s}_i, \mathbf{s}_j) = \sum_{k=1}^{d} \beta_k \sum_{l=1}^{L-k+1} \gamma_l \sum_{\substack{s=0 \\ s+l \leq L}}^{S(l)} \delta_S \mu_{k,l,s,\mathbf{s_i},\mathbf{s_j}} \tag{2.25}$$

$$\mu_{k,l,s,\mathbf{s_i},\mathbf{s_j}} = \mathbf{I}(\mathbf{u}_{k,l+s}(\mathbf{s}_i) = \mathbf{u}_{k,l}(\mathbf{s}_j)) + \mathbf{I}(\mathbf{u}_{k,l}(\mathbf{s}_i) = \mathbf{u}_{k,l+s}(\mathbf{s}_j)) \tag{2.26}$$

where β_k is the same as before; γ_l is the weights over the position l, SVM learns γ_l during the training process; $\delta_s = 1/(2(s+1))$ is the weight assigned to shifts of distance s, where $0 \leq s \leq S(l)$ subject to $s + l \leq L$; and $S(l)$ determines the shift range over the position l. $\mu_{k,l,s,\mathbf{s_i},\mathbf{s_j}}$ is the extended function which compare each base in extent of shifts s.

2.3 Feature Selection Methods

Feature selection methods are used to select the most informative features with respect to a prediction or classification problem. Eliminating redundant or uninformative features helps to enhance the generalization capability of machine learning algorithms and to improve the model interpretability. In our study, we used two feature selection methods: (1) SVM feature importance[GWBV02] and (2) information gain[XJK01], to identify the most relevant features for distinguishing alternatively spliced exons from constitutive exons. The weight vector $\mathbf{w} = \{|w_0|, |w_1|, ..., |w_n|\}$ (where n is the dimension of the feature vector) determined by the SVM algorithm is used as a heuristic to identify important features using the SVM feature importance method.

The information gain criterion also provides a simple way to determine feature importance. The information gain is the expected reduction in entropy caused by partitioning the training examples into classes, according to a certain feature (where the entropy measures the impurity of a sample E of training examples). One can rank all features in the order of decreasing information gain and select relevant features conservatively[XJK01]. A more robust way of identifying important features is to use a decision tree algorithm, which iteratively selects the feature with the highest information gain at each node of the tree. The features that are nodes in the final decision tree are considered to be more informative than the others.

Chapter 3

Data Set and Feature Construction

3.1 Data Set

3.1.1 *Tribolium Castaneum* EST Data Set

The use of *Tribolium castaneum* for developing our methods. We have used data from the red flour beetle (*Tribolium castaneum*) for developing and testing algorithms and tools for identifying and analyzing alternative splicing in insects. *Tribolium* has become the second most powerful arthropod (after *Drosophila*) for genetic and molecular genetic studies. It has a genome sequence of approximately 200 Mbp in 10 chromosomes. The genome sequence is well established; the third assembled version was released recently by the Human Genome Research Center of Baylor College of Medicine. In addition, approximately 56,000 ESTs (most contributed by scientists from KSU and the USDA ARS GMPRC,) from five stage-specific or tissue-enriched cDNA libraries are available for *Tribolium* (namely, adult hindgut and Malpighian tubules; ovary; adult head; larval carcass, including fat body; and mixed-stage, whole larvae).Previous studies have shown the importance of alternative splicing in *Tribolium*. For instance, *TcCHS1* chitin synthase[AHZ+04] and *TcLac2* laccase-2 genes[AMB+05] are found that these genes are alternatively spliced. They have compared the alternative splicing events for the same enzymes in several species.

With the EST data sets, we will design our experiments during which data sets will be cleaned, specifically, removing redundant ESTs. Then, we will align this cleaned EST data sets to the genomic sequences of *T. castaneum* by using alignment tools. And then the algorithm of identifying AS events will be implemented

3.1.2 Alternative Splicing Data Set for Recognition

The data set used in our SVM prediction experiments contains alternatively spliced and constitutive exons in *C.elegans*. The methods to derive this data set has been introduced in section 2.1. It has been used in related work[RSS05a] and is available at `http://www.fml.tuebingen.mpg.de/raetsch/projects/RASE`. A detailed description of how this data set was generated can be found in the wrok[RSS05a]. Briefly, *C.elegans* EST and full length cDNA sequences were aligned against the *C.elegans* genomic DNA to find the coordinates of exons and their flanking introns. After finding these coordinates, pairs of sequences which shared 3' and 5' boundaries of upstream and downstream exons were identified, such that one sequence contained an internal exon, while the other did not contain that exon. This procedure resulted in 487 alternatively spliced exons and 2531 constitutive exons. The final data set was split into 5 independent subsets of training and testing files for cross validation purposes.

3.2 Feature Construction

Six classes of features that affect alternative splicing are considered in our study: (1) pre-mRNA splicing motifs, specifically (1a) motifs derived from local sequences using MAST (MAST) and (1b) intronic regulatory splicing (IRS) motifs derived using comparative genomics methods; (2) pre-RNA secondary structure related features, specifically (2a) the optimal folding energy (OFE) and (2b) a reduced motif set (RMS) obtained by taking the secondary structures into account; (3) exon splicing enhancers (ESE); (4) splice site strength (SSS); (5) GC-content (GCC) in introns; and (6) basic sequence features (BSF) used in the work[RSS05a], specifically exon and flanking introns lengths and stop codon frames.

3.2.1 Splicing Motifs

We used the MEME[BWML06] and MAST[BG98] tools available at http://meme.sdsc.edu/meme/intro.html to detect motifs based on local sequences. MEME is a statistical tool for discovering *unknown* motifs in a group of related DNA or protein sequences. Its underlying algorithm is an extension of the expectation maximization algorithm for fitting finite mixture models[BE94]. Optimal values for parameters such as the motif width and the number of motif occurrences are automatically found by MEME. Contrary to MEME, MAST is a tool for searching sequences with a group of *known* motifs. A match score is calculated between each input sequence and each given motif. To use the MEME/MAST system, we first constructed local sequences by considering (-100, +100) bases around the donor sites (splice sites of upstream introns) and acceptor sites (splice sites of downstream introns) of the sequences in the original data set. Then, we ran MEME to obtain a list of 40 motifs (20 motifs for donor sites and 20 motifs for acceptor sites). MAST was used to search each sequence with these 40 motifs to obtain their location in each sequence and the corresponding p-values. Finally, we represented each sequence as a 40-dimensional feature vector. Each dimension corresponds to one of the 40 MEME motifs and indicates how many times that specific motif appears in the sequence.

In addition to motifs identified by MEME/MAST based on local sequences, we also considered intronic regulatory (IRM) motifs found by comparative genomics in Nematodes[KBSM+06]. The basic idea of the comparative genomics procedure here is to identify alternatively spliced exons whose flanking introns exhibit high nucleotide conservation between *C.elegans* and *C.briggsae*. Then, the most frequent pentamers and hexamers are extracted from the conserved introns. In our case, this procedure resulted in a list of 60 intronic regulatory motifs, 30 motifs for upstream introns and 30 motifs for downstream introns. For each sequence, we scanned the upstream intron with the upstream intronic motifs to find the number of occurrences of each motif. Each upstream intron is represented as a 30-dimensional vector, where each dimension indicates how many times the motif appears in the sequence. The same approach is applied to the downstream introns of each exons. Altogether, this set of features is represented as a 60-dimensional vector.

3.2.2 Structural Konwledge

It is known that the splicing of exons can be enhanced or repressed by specific local pre-mRNA secondary structures around the splice sites[HZBS07,PYR02]. As shown in previous work[HZBS07], motifs in single-stranded regions have more effect on the selection of splice sites than those in double-stranded regions. Following these ideas, we used the `mfold` software[MSZT99] available at `http://mfold.bioinfo.rpi.edu/` to predict the pre-mRNA folding (secondary structure formation) within a 100-base window around the acceptor and donor sites of each exon. `Mfold` parameters were chosen to prevent the formation of global double stranded base pairs. Thus, rather than folding the whole sequence, only local foldings were allowed. Two sub-classes of features related to the pre-mRNA secondary structure were considered in our study: (a) The *Optimal Folding Energy*, which roughly reflects the stability of the RNA folding; and (b) A *reduced motif set* derived, under the assumption that motifs on single stranded sequences are more effective than those on helices, from the set of MAST motifs by removing the motifs that are located on double stranded sequences with a probability greater than a threshold.

3.2.3 Splicing Regulators

Although splicing regulators have been identified in both introns and exons, exon splicing regulators (ESR) are more common and better characterized than intron splicing regulators[CCK02]. Exon splicing enhancers (ESE) affect the choice of splicing sites through recruiting arginine/serine dipeptide-rich (SR) proteins, which in turn bind other spliceosomal components through protein-protein interactions. We adopted the approach in the work[PMS07] to search for specific ESEs in our data. Since recent studies show that ESEs tend to be less active outside the close vicinity of splice sites[PMS07], we used a 50-base window around the splice sites to search for ESEs. We also considered the following two assumptions made in the RESCIE-ESE algorithm[FYSB02,PMS07] in our search: (1) ESEs appear much more frequently in exons than in introns and (2) ESEs appear much more frequently in exons with weak splice sites than in exons with strong splice sites. The following two difference distributions were computed in our study: (1) $\{|f_E^h - f_I^h||h \in \text{all possible hexamers}\}$, where f_E^h is the frequency of a given hexamer h in exon regions within the 50-base windows, and f_I^h is the frequency of a given hexamer h in intron regions; (2) $\{|f_W^h - f_S^h||h \in \text{all possible hexamers}\}$, where f_W^h is the frequency of a given hexamer in exons with weak splice sites, and f_S^h is the frequency of a given hexamer in exons with strong splice sites. Given these two difference distributions, we set a threshold and obtained 77 hexamers with high frequency in the two difference distributions. We scan the exon of each sequence for these motifs and represent the sequence as a 77-dimensional vector, where each dimension indicates how many times the corresponding hexamer appears in the sequence.

3.2.4 Characteristics of Splicing Sites

Another feature we used in our study is given by the strength of the splice sites, as the strength has been shown to be informative for identifying alternatively spliced exons[TS03,WM06]. More precisely, the strength is expected to be lower for alternatively spliced sites compared to constitutive splice sites. We used a position specific scoring-based approach[FH07] to model the strength of splice sites, according to the following formula: $score = \sum_i log\frac{F(X_i)}{F(X)}$, where $F(X_i)$ is the frequency of the nucleotide X at position i, and $F(X)$ is the background frequency of the nucleotide X. As already known, in *C.elegans* the background frequency is 66% AT. We extracted a range of (-3, +7) around donor sites (3 exon bases and 7 intron bases) and a range of (-26, +2) around acceptor sites (26 intron bases and 2 exon bases), and used the formula above to obtain scores for the strength of the acceptor and donor sites. The two ranges above are chosen to cover the main AG dinucleotides, which are bound by splicing factors around acceptor sites and the adjacent polypyrimidine tracts (PPT)[WM06]. Because the acceptor and donor sites can be seen as a pair, their scores are summed together to obtain the overall splice site strength, which is represented as another feature.

3.2.5 Sequence Features

The GC content of a sequence is another feature correlated with the selection of splice sites. Alternatively spliced exons occur more frequently in GC-poor flanking sequences[TS03]. We take into account this property by using a sliding window method to scan the GC content of each sequence within a range of (+100, -100) around donor and acceptor sites. The window size is set to 5, resulting in a 40-dimensional feature vector for each splice site. Each position indicates the ratio of GCs to the window size.

Last but not the least, sequence length has been shown to be a feature that can help distinguish alternatively spliced exons from constitutive exons[SSC+04,DSS05b]. In[RSS05a], a feature vector consisting of upstream intron length, exon length, downstream intron length and the frame of the stop codon was constructed for each exon and its flanking introns. The length features were discretized into a logarithmically spaced vector consisting of 30 bins. The frame of the stop codons is represented using a 3D vector. In this study, we call this last set of features *basic features*.

Chapter 4

Experimental Results

4.1 EST-based Analysis of Alternative Splicing in *Tribolium.*

Pipeline for EST data analysis.

EST data analysis is an essential first step for all EST projects. Several EST data analysis pipelines are available as web-servers, e.g. ESTpass[LHB+07], EGassembler[MNTK+06] and ESTexplorer[NDGR07]. However, they all have limitations, e.g., with respect to the amount of data that can be uploaded at once or the type and format of the annotations and statistics they provide. Given the increasing number of EST projects at KSU and the limitations of the publicly available pipelines, we have developed a local ArthropodEST pipeline for EST analysis ([CPK+07] http://129.130.115.231/www_est/i3.html within the KSU domain), using existing open source software tools. A prototype of browsable ArthropodEST database has been created and will be refined in the near future. Other tools for annotation will be added to the pipeline and customized statistics will be available.

Alternative splicing analysis in *Tribolium*.

We have taken the first steps towards the analysis of alternative splicing in *Tribolium*. We have used the traditional approach to identifying alternative splicing, that is, aligning ESTs to the genome using GMAP[WW05] and Exonerate[SB05]. So far, we have analyzed GMAP results using ASpipe, an alternative splicing analysis tool developed by[WOBY08]. ASpipe analysis finds five types of alternative splicing events: alternative donor (**AltD**), alternative acceptor (**AltA**), alternative donor and acceptor (**AltP**), exon skipping (**ExS**) and intron retention (**IntR**). A total of 357 events are found in 213 *Tribolium* genes based on our EST data. Our preliminary analysis (Table 4.1) shows that intron retention is the most common type of alternative slicing event in *Tribolium*, followed by exon skipping.

Total	AltD	AltD%	AltA	AltA%	AltP	AltP%	ExS	ExS%	IntR	IntR%
357	45	13%	40	11%	27	7%	109	31%	136	38%

Table 4.1: *Number of occurrences and percentage for each alternative splicing event.*

ASpipe found 5809 expressed genes and estimated that approximately 4% of the genes in *Tribolium* undergo alternative splicing. Given that we have a relatively small number of ESTs, we expect that in reality many more *Tribolium* genes are alternatively spliced. Cross-species EST to genome alignments and tiling arrays are expected to improve this initial estimate. The results of the ASpipe analysis have been displayed using ASview[WOBY08]. The alternative events are marked in the viewer and are linked to the actual alignments, so that one could easily judge the correctness of the events found. Tools like ASview will prove invaluable for the manual analysis necessary to validate results of this bioinformatics approach. A preliminary examination of the results using ASview showed interesting splicing events. A more careful examination is needed to draw definite conclusions.

4.2 Alternative Splicing Prediction suing SVM

4.2.1 Motif Evaluation

The purpose of the motif evaluation in this section is to identify the splicing motifs that appear in several different sets, as those motifs are probably the most informative for alternative splicing. To do that, we first compared the set of 40 motifs identified by MEME/MAST with the set of putative motifs found in[RSS05a] and the ISR motifs found in[KBSM+06]. The MAST motifs are represented as position-specific scoring matrices (PSSMs), shown as a

Table 4.2: *Intersection between MAST motifs, motifs found in*[RSS05a] *and ISR motifs found in*[KBSM+06]*. MAST motifs 1-20 are around 5' splice sites, while motifs 21-40 are around 3' splice sites. ISR motifs are italicized.*

MAST motifs	E-value	Contained	Number
(Multilevel expression)		hexamers	
TTTTTTTTTTCA	4.8e-046	tttttt	mast2
GTGAGTTTTTT A	4.6e-033	tttttt	mast3
AAAAATTTTAAATTTTCAGG TT TTAAAATTT A	3.9e-030	tttttt, atatat tatata	mast4
ATTTTTCAAATTTTT T C T A C	1.6e-026	tttttt	mast6
GCCGGTGGAGCTGTCGTAGG A A CC CC GC GTAGC A	3.6e-026	gttgtc, *catcgc* *gtgttg*	mast9
AGCCGCCGAAGCCCTTGCCA CATT TA C AAAGCC GAG	1.0e-018	gttgtc , *ccctgg* *catcgc, cactgc*	mast14
CAGCACCAACAGCACCACCA TC TG G TT G A	1.4e-049	cagcag	mast22
TTTTTTTTTTCAAAATTTTA A TGG T CT	3.3e-038	tttaaa, aatttt atttta	mast23

two-level consensus sequences in Table 4.2. Upper-level bases have scores higher than or equal to the lower-level bases. A base is conserved if there is no lower-level base in its column. Eight motifs are found in all three sets compared, some of them (e.g., mast2 and mast3) being highly conserved among the *C.elegans* sequences in our data set.

Second, we compared the 77 ESE hexamers, found as described in Section 3.2.3., with two sets of candidate human and mouse ESE hexamers proposed in[RES]. Thirty two out of the 77 putative *C.elegans* ESE hexamers occur also in the human and mouse ESE sets, suggesting that the regulation of splicing, as well as the splicing process itself, are highly conserved in metazoans. Furthermore, a set of experimentally confirmed A. *thaliana* ESE ninemers[PMS07] was used for comparison. The 32 conserved ESE hexamers are shown below; the A. *thaliana* ESE ninemers containing some of these hexamers are listed in brackets:

aatgga, aacaac, **aagaag** [GAAGAAGAA, GAGAAGAAG, TTGAAGAAG], **aaggaa** [GAAGGAAGA], **aaggag** [AAAGGAGAT], attgga, atgatg, atggaa, atggat, acaaga, **agaaga** [GAAGAAGAA, GAGAAGAAG], agaagc, tcatca, tgaaga, tgatga, tggaag, tggatc, **caagaa** [CAAGAAACA], **cagaag** [GAGCAGAAG], cgacga, gaaagc, **gaagaa** [GAAGAAGAA, GAGAAGAAG, GAAGAAAGA, TTGAAGAAG], **gaagat** [GAAGATGGA, GAAGATTGA], **gaagag** [GAAGAGAAA], **gaagga** [GAAGGAAGA], gatgat, **gatgga** [GAAGATGGA], gagaag, gaggag, ggaaga [GAAGGAAGA], **ggagaa** [ATGGAGAAA], ggagga.

It is worth mentioning that our study finds no intersection between the ISR motifs and the ESE motifs in *C.elegans*, suggesting that the two sets are functionally different.

4.2.2 Model Selection

The performance of a classifier depends on judicious choice of various parameters of the algorithm. For the SVM algorithm there are several inputs that can be varied: the cost of constraint violation C (e.g., $C = 1$), tolerance of the termination criterion (e.g., $\epsilon = 0.01$), type of kernel used (e.g., linear, polynomial, radial or Gaussian), parameters of the kernel (e.g., the degree or coefficients of the polynomial kernel), etc.

Table 4.3: *Results of alternatively spliced exons classification. All features, but ISR motifs, are included.*

	C	Validation Score		Test score	
		fp 1%	AUC	fp 1%	AUC
Split1	0.05	35.36%	86.99%	44.44%	89.32%
Split2	0.05	36.50%	88.56%	46.92%	87.57%
Split3	0.1	35.27%	86.91%	47.31%	88.59%
Split4	0.01	37.56%	88.36%	26.88%	86.60%
Split5	0.1	39.80%	88.03%	29.47%	86.98%

G. Rätsch et al.[RSS05a] have used basic features with several types of customized kernels, as well as an optimal sub-kernel weighting to learn SVM classifiers that differentiate between alternatively spliced and constitutive exons, and to identify motifs that can be used to interpret the results. In this section, we show that simple linear kernels can be used to obtain similar results if motifs are used as input features. In order to tune the cost C, we use 5-fold cross-validation for each training set, with $C \in \{0.01, 0.05, 0.1, 0.5, 1, 2\}$. We choose the value of C for which the area under curve (AUC) is maximized during the cross-validation. AUC is a global measurement which takes true positive ratio and false positive ratio into account. True positive ratio is the number of positively labeled examples classified by the algorithm as positive divided by the total number of positive examples. False positive ratio is the number of negatively labeled examples classified as positive divided by the number of negatively labeled examples.

Table 4.3 shows the results of classification of exons using all features described in Section 3, except conserved ISR motifs that need additional information from closely related organisms to be determined. Table 4.4 shows the results when the conserved ISR motifs described in Section 3.2 are also included.

From Tables 4.3 and 4.4, we notice that on the average, the performance improves in terms of true positive rate at 1% false positive rate when ISR motifs are included, which means that ISR motifs conserved among several species contribute to better classification performance. Furthermore, the results are comparable and sometimes better than the results

Table 4.4: *Results of alternatively spliced exons classification. All features, including ISR motifs are used.*

	C	Validation Score		Test score	
		fp 1%	AUC	fp 1%	AUC
Split1	0.05	32.45%	86.55%	56.48%	90.05%
Split2	0.05	39.33%	88.32%	52.04%	89.04%
Split3	0.1	37.56%	87.76%	38.71%	87.97%
Split4	0.01	40.86%	89.02%	37.63%	84.42%
Split5	0.1	36.48%	87.50%	35.79%	85.69%

obtained by G. Rätsch et al.[RSS05a]. For example, when testing on the first data set we obtain a true positive rate of 56.48% at a fp rate of 1% and the AUC is 90.05%, thus improving the previous results of tp 51.85% at fp 1% and AUC 89.90%.

To evaluate how much the mixed features improve the performance of classification of alternatively spliced exons, we compared the AUC scores of classifiers trained on data sets with and without mixed features, respectively. Figure 4.1 shows the result of comparison between a data set with basic features only and a data set that includes the other features (except conserved ISR motifs).

Figure 4.2 shows a comparison of the AUC scores for each data set. It can be seen that the SVM classifiers using MAST motif features return higher AUC scores than those considering only basic sequence features.

In order to evaluate the effect of pre-mRNA secondary structure features on classification of alternatively spliced exons, we performed two experiments, one using data sets considering pre-mRNA secondary structure features obtained as described in Section 3.2 and the other using data sets without secondary structure features. Figure 4.3 shows the results of the two experiments in which the classifiers were trained using 5-fold cross-validation with optimal cost parameters listed in Table 4.3. We can see the improvement obtained when considering secondary structure features.

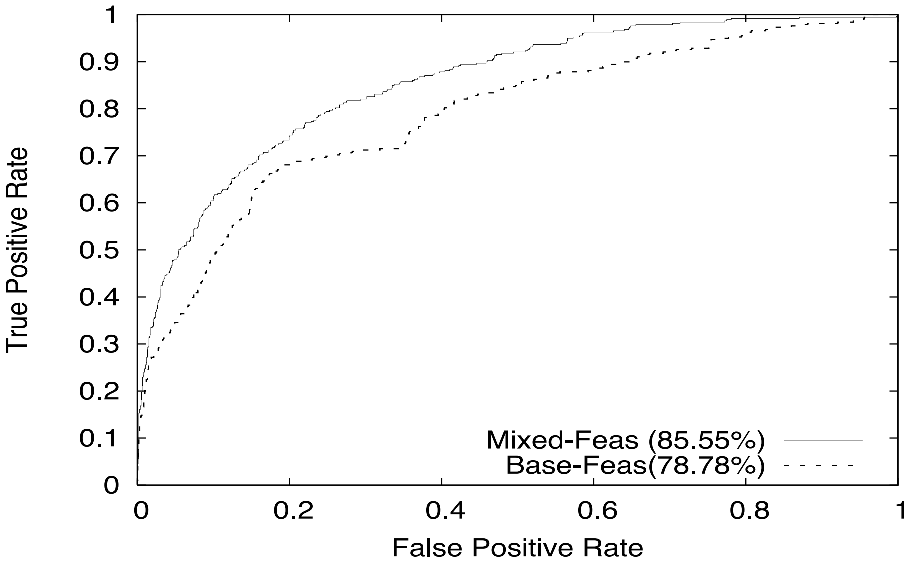


Figure 4.1: *Comparison of ROC curves obtained using basic features only and basic features plus other mixed features (except conserved ISR motifs). Models trained by 5-fold CV with* $C = 1$.

4.3 Feature Selection

We used SVM feature importance and information gain criteria to order features according to their importance with respect to the problem of predicting alternatively spliced exons. First, a linear kernel SVM classifier with optimal cost value was learned for each dataset. The importance of each class of features was estimated by taking the average, across all features in a class, of the corresponding feature weight in the weight vector **w**. Table 4.5 shows the statistics obtained for the classes of features considered. It can be seen that SSS and BSF are the most informative classes of features. It is not surprising that these classes of features have high importance, as they were previously reported to be very informative for exon splicing prediction in[WM06] and[SSC+04], respectively. However, taken separately, the SSS features do not discriminate well between alternatively spliced and constitutive exons (results not shown), suggesting that they are highly correlated with the BSF features.

In Section 4.2., we have seen that ISR motifs, MAST motifs and ESEs provide useful information for classification, improving the results of classifiers that use only BSF and SSS

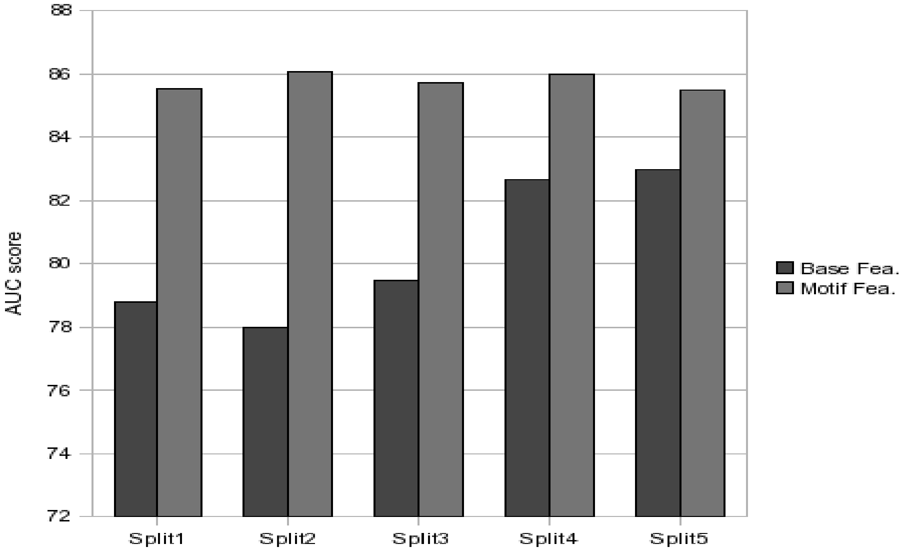


Figure 4.2: *AUC score comparison between data sets with BSF and data sets including MAST motif features. AUC values were obtained based on 5-fold CV with* $C = 1$.

Table 4.5: *Weight importance of the following features: 105 BSF, 1 SSS, 80 GCC, 60 ISR, 40 MAST, 77 ESE, 1 OFE.*

Feature	Mean	Max	Min	Std. Dev.
BSF	16.61	27.48	0.13	9.87
SSS	51.05	51.05	51.05	0.00
GCC	10.60	14.90	6.65	1.77
ISR	10.14	25.93	3.43	4.41
MAST	2.06	3.80	0.27	1.02
ESE	1.08	2.13	0.45	0.32
OFE	0.18	0.12	0.24	0.06

features. To select the most informative motifs from these sets of features, we used the SVM-produced weight value to order the motifs and chose the best 20 motifs among these features. Most of the 20 best motifs were ISR motifs.

Furthermore, as described in Section 3.3, we also ran the J48 decision tree algorithm in the data mining package WEKA[WF05] to build a classifier for each data set. We analyzed the nodes in each constructed decision tree and extracted the motifs, namely nodes, occurring in all five trees. We consider these motifs as most informative motifs according to the

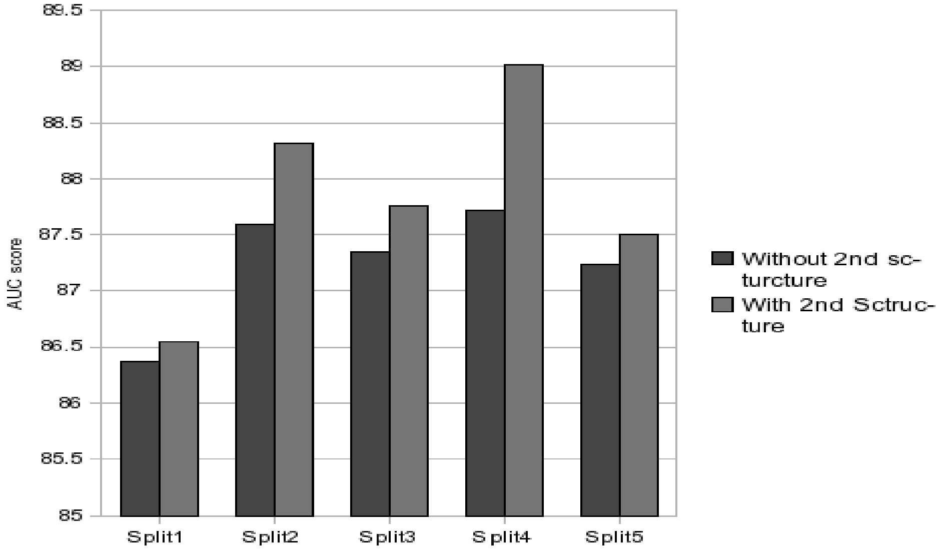


Figure 4.3: *AUC scores comparison between data sets with features of secondary structure and data sets without features of secondary structure*

information gain criterion. Table 4.6 shows the list of motifs found based on information gain. By comparing the set of the 20 best SVM motifs with the set of the best J48 motifs, we found that the ISR pentamers GCTTC and GTGTG in the upstream intron and GCATG in the downstream intron were included in both sets (bolded in Table 4.6). We also noted that ese65 (gatgat) was the most frequent hexamer among the selected ESEs.

Table 4.6: *List of mastk, esek and irsk motifs found by choosing nodes which occur in all decision tree classifiers, where k indicates the position in the corresponding list. Irs21,23,31 are ISR motifs identified by both J48 and SVM as important. The rank is based on SVM feature importance.*

motifs	Location	Weight value	Rank
mast4	5' ss	1.59	272
mast17	5' ss	2.73	245
mast22	3' ss	3.35	238
mast23	3' ss	3.33	240
mast32	3' ss	1.34	283
ese20	5' ss	1.23	288
ese65	3 ' ss	1.85	262
irs7	5' intron	6.15	217
irs9	5' intron	10.18	134
irs14	5' intron	10.39	132
irs21	5' intron	16.05	62
irs23	5' intron	13.52	75
irs31	3' intron	11.76	109
irs49	3' intron	10.06	135

Chapter 5

Conclusions and Future Work

The importance of identifying alternative splicing informative features and using them to predict alternative splicing events is reflected by the amount of recent research in this area[DSS05b,RSS05a,SA03,SSC+04]. However, there is no comprehensive computational study that considers all the features that have been shown experimentally to contribute to the identification of alternatively spliced exons. In this thesis, we have presented such a study.

More precisely, we have shown how to use computational methods to construct alternative splicing features and how to built simple SVM classifiers using the features constructed. Our ultimate goal was to gain insights into the most informative features for the prediction problem at hand. MEME/MAST tools were used to identify motifs from local sequences. We have demonstrated that the resulting motifs can aid the classification of alternatively spliced exons even when used with simple linear SVM classifiers, thus providing a good alternative to more sophisticated kernel methods[RSS05a]. We have also explored several other features, such as pre-mRNA secondary structure, exonic splicing enhancers, splice site strength and CG-content, which have been shown to be relevant to alternative splicing from a biological point of view. Our results indicate that these features can further improve the accuracy of classifiers that distinguish alternatively and constitutively spliced exons. Finally, we have shown how we can use features selection methods to identify informative features. The methods presented here will be useful for the analysis of predicted gene models in newly sequenced genomes with limited, but enough for training, ESTs and/or cDNA libraries.

Our future work will focus on 1) identifying motifs more accurately. 2) we will also explore alternative ways to represent biological features, as well as relationships among biological features (e.g., pre-mRNA secondary structures and motifs) or between biological features and environment. 3) we will explore to rerunn the approaches in previous work and futher to incoporate our feature sets into previous methods, which is aimed to increasing the accuracy of identifying AS events. 5) We can adapt this approach in this thesis work to other organism which does not have limited EST coverage. 6) we will design new algorithms to predict not only the exon-skipping type of AS, but also other types of AS.

Bibliography

[AHZ$^+$04] Y. Arakane, DG. Hogenkamp, YC. Zhu, KJ. Kramer, CA. Specht, RW. Beeman, MR. Kanost, and S. Muthukrishnan, *Characterization of two chitin synthase genes of the red flour beetle, tribolium castaneum, and alternate exon usage in one of the genes during development.*, Insect Biochem Mol Biol. **34** (2004), no. 3, 291–304.

[AMB$^+$05] Y. Arakane, S. Muthukrishnan, RW. Beeman, MR. Kanost, and KJ. Kramer, *Laccase 2 is the phenoloxidase gene required for beetle cuticle tanning.*, PNAS **102** (2005), no. 32, 11337–11342.

[AMM44] O. T. Avery, C. M. MacLeod, and M. McCarty, *Studies on the Chemical Nature of the Substance Inducing Transformation of Pneumococcal Types: Induction of Transformation by a Desoxyribonucleic Acid Fraction Isolated from Pneumococcus Type III*, Journal of Experimental Medicine **79** (1944), no. 1, 137–158.

[Ast04] G. Ast, *How did alternative splicing evolve?*, Nat. Rev. Genet. **5** (2004), no. 10, 773–782.

[BE94] Timothy L. Bailey and Charles Elkan, *Fitting a mixture model by expectation maximization to discover motifs in biopolymers*, Proc. of 2nd International Conf. on Intelligent Systems for Molecular Biology (1994), 28–36.

[BG98] Timothy L. Bailey and Michael Gribskov, *Combining evidence using p-values: application to sequence homology searches*, Bioinformatics **14** (1998), no. 1, 48–54.

[BHB03] Asa Ben-Hur and Douglas Brutlag, *Remote homology detection: a motif based approach*, Bioinformatics **19** (2003), no. Suppl. 1, i26–i23.

[Bre74] S. Brenner, *The genetics of caenorhabditis elegans*, Genetics **77** (1974), 71–94.

[BRP06] Paola Bonizzoni, Raffaella, and Graziano Pesole, *Computational methods for alternative splicing prediction*, Brief Funct Genomic Proteomic **5** (2006), no. 1, 46–51.

[BSt] Peter Bartlett and John Shawe-taylor, *Generic author design sample pages 1998/04/10 13:50 1 generalization performance of support vector machines and other pattern classifiers.*

[Bur98] Christopher J. C. Burges, *A Tutorial on Support Vector Machines for Pattern Recognition*, Data Mining and Knowledge Discovery **2** (1998), no. 2, 121–167.

[BWML06] Timothy L. Bailey, Nadya Williams, Chris Misleh, and Wilfred W. Li, *MEME: discovering and analyzing DNA and protein sequence motifs*, Nucleic Acids Research **34(Web Server issue)** (2006), W369W373.

[CCK02] Luca Cartegni, Shern L. Chew, and Adrian R. Krainer, *Listening to silence and understanding nonsense: Exonic mutations that affect splicing*, Nature reviews Genetics **3** (2002), no. 4, 285–298.

[CPK+07] S. Chellapilla, Y. Park, S. Kallumadi, S.J. Brown, and D. Caragea, *Arthropodest: Software pipeline for arthropod est data analysis, bioinformatics center, ksu*, Nucleic Acids Research (2007).

[DEH+07] Hattori D., Demir E., Kim HW, Viragh E, Zipursky SL, and Dickson BJ, *Dscam diversity is essential for neuronal wiring and self-recognition.*, Nature **449** (2007), no. 7159, 223–7.

[DSS05a] Gideon Dror, Rotem Sorek, and Ron Shamir, *Accurate identification of alternatively spliced exons using support vector machine*, Bioinformatics **21** (2005), no. 7, 897–901.

[DSS05b] ———, *Accurate identification of alternatively spliced exons using support vector machine*, Bioinformatics **21** (2005), no. 7, 897–901.

[FH07] Marie E. Fahey and Desmond G. Higgins, *Gene Expression, Intron Density and Splice Site Strength in Drosophila and Caenorhabditis*, Journal of Molecular Evolution. **65** (2007), no. 3, 349–357.

[FYSB02] William G. Fairbrother, Ru-Fang Yeh, Phillip A. Sharp, and Christopher B. Burge, *Predictive identification of exonic splicing enhancer motifs in human protein-coding genes*, Science **297** (2002), no. 5583, 1007–1013.

[Gil78] W. Gilbert, *Why genes in pieces?*, Nature **271** (1978), no. 5645, 501.

[Gra01] B. Graveley, *Alternative splicing: increasing diversity in the proteomic world*, Trends Genet. **17** (2001), no. 2, 100–107.

[GWBV02] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, *Gene selection for cancer classification using support vector machines*, Machine Learning **46** (2002), 389–422.

[HO08] Dirk Holste and Uwe Ohle, *Strategies for Identifying RNA Splicing Regulatory Motifs and Predicting Alternative Splicing Events*, PLoS Comput Biol **4** (2008), no. 1, e21.

[HZBS07] Michael Hiller, Zhaiyi Zhang, Rolf Backofen, and Stefan Stamm, *Pre-mRNA Secondary Structures Influence Exon Recognition*, PLoS Comput Biol **3** (2007), no. 11, e204.

[int] *(The International Genome Sequencing Consortium, 2001) Initial sequencing and analysis of the human genome. Nature. 2001 Feb 15;409(6822):860-921.*

[JCGE$^+$03] J. M. Johnson, J. Castle, P. Garrett-Engele, Z. Kan, P. M. Loerch, C. D. Armour, R. Santos, E. E. Schadt, R. Stoughton, and D. D. Shoemaker, *Genome-wide survey of human alternative pre-mrna splicing with exon junction microarrays*, Science. **302** (2003), no. 5653, 2141–2144.

[KBSM$^+$06] Jennifer L. Kabat, Sergio Barberan-Soler, Paul McKenna, Hiram Clawson, Tracy Farrer, and Alan M. Zahler, *Intronic Alternative Splicing Regulators Identified by Comparative Genomics in Nematodes*, PLoS Comput Biol. **2** (2006), no. 7, e86.

[KRGS01] Zhengyan Kan, Eric C. Rouchka, Warren R. Gish, and David J. States, *Gene Structure Prediction and Alternative Splicing Analysis Using Genomically Aligned ESTs*, Genome Res. **11** (2001), no. 5, 889–900.

[KSG02] Z Kan, D States, and W Gish, *Selecting for functional alternative splices in ests*, Genome Res. **12** (2002), no. 12, 1837–45.

[KTB$^+$02] M. Koslowski, O. Türeci, C. Bell, P. Krause, HA. Lehr, J. Brunner, G. Seitz, F.O. Nestle, C. Huber, and Sahin U., *Multiple splice variants of lactate dehydrogenase c selectively expressed in human cancer.*, Cancer Research **62** (2002), no. 22, 6759–5.

[LEC$^+$03] Christina S. Leslie, Eleazar Eskin, Adiel Cohen, Jason Weston, and William Stafford Noble, *Mismatch string kernels for discriminative protein classification*, Bioinformatics **20** (2003), no. 4, 467–476.

[LHB$^+$07] B. Lee, T. Hong, SJ. Byun, T. Woo, and YJ. Choi, *Estpass: a web-based server for processing and annotating expressed sequence tag (est)*, Nucleic Acids Research **35** (2007), No. suppl 2 W159–W162.

[LTR04] B. T. K. Lee, T. W. Tan, and S. Ranganathan, *Dedb: a database of drosophila melanogaster exons in splicing graph form.*, BMC Bioinformatics. **5** (2004), 189.

[MNTK$^+$06] A. Masoudi-Nejad, K. Tonomura, S. Kawashima, Y. Moriya, M. Suzuki, M. Itoh, M. Kanehisa, T. Endo, and S. Goto, *Egassembler: online bioinformatics service for large-scale processing, clustering and assembling ests and genomic dna fragments.*, Nucleic Acids Research **34** (2006), W459–W46.

[MSZT99] David H. Mathews, Jeffrey Sabina, Michael Zuker, and Douglas H. Turner, *Expanded Sequence Dependence of Thermodynamic Parameters Improves Prediction of RNA Secondary Structure*, Journal of Molecular Biology **288** (1999), no. 5, 911–940.

[NDGR07] SH. Nagaraj, N. Deshpande, RB. Gasser, and S. Ranganathan, *Estexplorer: an automated assembly and annotation platform to analyse expressed sequence tags (ests).*, Nucleic Acids Research (2007).

[NGR06] Shivashankar H. Nagaraj, Robin B. Gasser, and Shoba Ranganathan, *A hitchhiker's guide to expressed sequence tag (EST) analysis*, Brief Bioinform. **8** (2006), no. 1, 6–21.

[Pea06] H. Pearson, *Genetics: what is a gene?*, Nature **441** (2006), no. 7092, 398–401.

[PMS07] Mihaela Pertea, Stephen M Mount, and Steven L Salzberg, *A computational survey of candidate exonic splicing enhancer motifs in the model plant arabidopsis thaliana*, BMC Bioinformatics **8** (2007), 159.

[PYR02] Donald J. Patterson, Ken Yasuhara, and Walter L. Ruzzo, *PRE-mRNA Secondary Structure Prediction Aids Splice Site Prediction*, Proceedings of the Pacific Symposium on Biocomputing (2002), 223–234.

[RES] *Rescue-ese web server*, [http://genes.mit.edu/burgelab/rescue-ese/].

[RS04] G. Rätsch and S. Sonnenburg, *Accurate Splice Site Dectection for Caenorhabditis Elegans in Kernel Methods in Computational Biology*, Kernel Methods in Computational Biology, MIT press (2004), 277–298.

[RSS05a] G. Rätsch, S. Sonnenburg, and B. Schölkof, *RASE: recognition of alternatively spliced exons in c. elegans*, Bioinformatics **21** (2005), no. Suppl 1, 369–377.

[RSS05b] G. Rätsch, S. Sonnenburg, and B. Schölkopf, *Rase: recognition of alternatively spliced exons in c.elegans*, Bioinformatics **21** (2005), no. 1, 369–377.

[SA03] Rotem Sorek and Gil Ast, *Intronic sequences flanking alternatively spliced exons are conserved between human and mouse*, Genome Research **13** (2003), no. 7, 1631–1637.

[SB05] GSC. Slater and E. Birney, *Automated generation of heuristics for biological sequence comparison.*, BMC Bioinformatics (2005), no. 6, 31.

[SCS+00] D. Schmucker, J. C. Clemens, H. Shu, C. A. Worby, J. Xiao, M. Muda, J. E. Dixon, and S. L. Zipursky, *Drosophila dscam is an axon guidance receptor exhibiting extraordinary molecular diversity.*, Cell **101** (2000), 671–684.

[SRJM02] S. Sonnenburg, G. Ratsch, A. Jagota, and K. Muller, *New methods for splice site recognition*, 2002.

[SRS05] Sören Sonnenburg, Gunnar Rätsch, and Christin Schäfer, *Learning interpretable svms for biological sequence classification*, RECOMB 2005, LNBI 3500 (Berlin Heidelberg), Springer-Verlag, 2005, pp. 389–407.

[SSC+04] Rotem Sorek, Ronen Shemesh, Yuval Cohen, Ortal Basechess, Gil Ast, and Ron Shamir, *A Non-EST based method for exon-skipping prediction*, Genome Res. **14** (2004), no. 8, 1617–1623.

[SSP$^+$07] Sören Sonnenburg, Gabriele Schweikert, Petra Philips, Jonas Behr, and Gunnar Rätsch, *Accurate splice site prediction using support vector machines*, BMC Bioinformatics **8** (2007), no. Suppl 10, S7.

[TS03] T. A. Thanaraj and S. Stamm, *Prediction and statistical analysis of alternatively spliced exons*, Prog Mol Subcell Biol. **31** (2003), 1–31.

[Vap99] Vladimir N. Vapnik, *The Nature of Statistical Learning Theory (Statistics for Engineering and Information Science).*, Springer Verlag, December 1999.

[ven] *(venter et al., 2001) the sequence of the human genome. science 16 february 2001:vol. 291. no. 5507, pp. 1304 1351.*

[WF05] Ian H. Witten and Eibe Frank, *Data Mining: Practical machine learning tools and techniques*, second ed., Morgan Kaufmann, San Francisco, 2005.

[WM06] Magnus Wang and Antonio Marin, *Characterization and prediction of alternative splice sites*, Gene **366** (2006), no. 2, 219–227.

[WOBY08] B.-B. Wang, M. O'Toole, V. Brendel, and N.D. Young, *Novel and conserved alternative splicing events are revealed by cross-species est alignments in legumes.*, BMC Plant Biol **8** (2008), no. 17.

[WW05] T. D. Wu and C. K. Watanabe, *Gmap: a genomic mapping and alignment program for mrna and est sequences.*, Bioinformatics (2005), no. 21, 1859–1875.

[XCB08] J. Xia, D. Caragea, and S. Brown, *Exploring alternative splicing features using support vector machine*, IEEE Bionformatics and Biomedicine (BIBM) (2008), 230–238.

[XJK01] Eric P. Xing, Michael I. Jordan, and Richard M. Karp, *Feature selection for* *high-dimensional genomic microarray data*, Proc. 18th International Conf. on Machine Learning (2001), 601–608.

[Zah05] A.M. Zahler, *Alternative splicing in c. elegans.*, WormBook, ed. The C. elegans Research Community, 2005.