ABSTRACT

Title of Dissertation:	FEATURE GENERATION AND ANALYSIS APPLIED TO SEQUENCE CLASSIFICATION FOR SPLICE-SITE PREDICTION
	Rezarta Islamaj Doctor of Philosophy, 2007
Dissertation Directed By:	Dr. Lise Getoor Department of Computer Science

Sequence classification is an important problem in many real-world applications. Sequence data often contain no explicit "signals," or features, to enable the construction of classification algorithms. Extracting and interpreting the most useful features is challenging, and hand construction of good features is the basis of many classification algorithms. In this thesis, I address this problem by developing a feature-generation algorithm (FGA). FGA is a scalable method for automatic feature generation for sequences; it identifies sequence components and uses domain knowledge, systematically constructs features, explores the space of possible features, and identifies the most useful ones.

In the domain of biological sequences, splice-sites are locations in DNA sequences that signal the boundaries between genetic information and intervening non-coding

regions. Only when splice-sites are identified with nucleotide precision can the genetic information be translated to produce functional proteins. In this thesis, I address this fundamental process by developing a highly accurate splice-site prediction model that employs our sequence feature-generation framework. The FGA model shows statistically significant improvements over state-of-the-art splice-site prediction methods.

So that biologists can understand and interpret the features FGA constructs, I developed SplicePort, a web-based tool for splice-site prediction and analysis. With SplicePort the user can explore the relevant features for splicing, and can obtain splice-site predictions for the sequences based on these features. For an experimental biologist trying to identify the critical sequence elements of splicing, SplicePort offers flexibility and a rich motif exploration functionality, which may help to significantly reduce the amount of experimentation needed. In this thesis, I present examples of the observed feature groups and describe efforts to detect biological signals that may be important for the splicing process.

Naturally, FGA can be generalized to other biologically inspired classification problems, such as tissue-specific regulatory elements, polyadenylation sites, promoters, as well as other sequence classification problems, provided we have sufficient knowledge of the new domain.

FEATURE GENERATION AND ANALYSIS APPLIED TO SEQUENCE CLASSIFICATION FOR SPLICE-SITE PREDICTION

By

Rezarta Islamaj

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2007

Advisory Committee:
Assistant Professor Lise Getoor, Chair
Associate Professor Stephen M. Mount
Doctor W. John Wilbur
Professor Samir Khuller
Associate Professor Chau-Wen Tseng

Dedication

To my parents.

Acknowledgements

Upon accomplishing a big project everyone has someone to be thankful for, someone that has offered support and given advice, being this in the form of thoroughly important criticism or plain and simple love. I am very fortunate to have a whole team.

My advisor, Lise Getoor, has not only encouraged me to work on a project of my choice, but helped me develop it into a research problem that is relevant, challenging and interesting. She is definitely a role model for me in every aspect of my life and I most sincerely cannot thank her enough.

My committee: Stephen M. Mount, W. John Wilbur, Samir Khuller and Chau-Wen Tseng, has followed my progress with interest and provided me with invaluable insight and suggestions in my research and experiments. It was a pleasure working with them in every aspect.

Especially, I would like to thank Dr. Stephen Mount for his continued support, strong encouragement and useful guidance. During the collaboration with him, I came to realize the importance of cross-disciplinary research and how the two fields can mutually depend and help one another. I am very fortunate to have known him and worked with him and I would like to express my profound gratitude to Dr. Mount for challenging me to become a better scientist.

Moreover, Dr. John Wilbur has very generously helped and supported me for all my years as a part of his research group at NCBI. The NCBI pre-doctoral fellowship, granted by ORISE and supervised by him, allowed me to access not only the resources and the coding infrastructure of the Computational Biology Branch and the IRET group, but also a constant interaction with brilliant scientists such as Dr. John Wilbur himself.

I also want to express my thanks to Professor Samir Khuller for attentively listening to me five years ago, and not only sympathizing with me for wanting to do research in bioinformatics, but for directing me to the NCBI internship opportunity that came up, which transformed into my thesis research.

I want to thank my friends, Indrajit Bhattacharja, Mustafa Bilgic, Louis Licamele, Galileo Namata, Elena Zheleva, Prithviraj Sen, Tugba Onal Suzek, Baris Suzek, Burcu Karagol-Ayan, Fusun Yaman Sirin, Elena Zotenko, Adam Lee, Lana Yeganova, Svetlana Shabalina and others…

This dissertation could not have been possible without the love and support of my dear parents, Hariklina and Islam Islamaj, who made my education their top priority and lead me to where I am right now. My father has always been a true inspiration of everything true and just. He gave me the principles of work and discipline in little, but solid incremental steps. He taught me to be thorough in my work and always believe in myself.

I never realized just how much my mother had taught me, until the day I became a mother. She is a true inspiration for everything good and sweet. She taught me to believe in family values and gave me the principles of economizing and parallel processing long before I studied the theory. She was unbelievable at managing a working career and a family life in a most remarkable manner. I am proud of being their daughter.

There are no words that could express my deepest gratitude to my brother Ergys and my husband Gunay. Ergys, for being the absolutely most wonderful brother anyone can wish for, for being there for me every single time I needed a shoulder, a hand, or a word, for simply being my very best friend, my very best brother, my daughter's very best uncle and my husband's very best friend. And Gunay, for his love and company, his care and patience, his logic and intelligence, his steadiness and stubbornness into getting things done in the most accurate and rightful manner. We started this journey together and it has been very memorable every single moment of it. Both of them have been for me, one, the rock that always reminds me where our roots are, and the other, the eagle, that always tells me that it doesn't matter how high we fly, as long as we do it with dignity and integrity.

And finally, I want to mention my precious daughter, Ilayda, for making my life so challenging and so rewarding, for inspiring me every day, for hugging me every night, and for making all the difference in my life.

.

Table of Contents

List of Figures and Tables

Chapter 1: Introduction

Many data-mining problems involve data that is best represented as a sequence. Sequence data comes in many forms, including 1) human communication, such as speech, handwriting, and printed text; 2) time series, such as stock prices, temperature readings, and web-click streams; and 3) biological sequences, such as DNA, RNA, and proteins. Sequence data in these domains may exhibit certain characteristics and relationships. Let us consider, for example, a document written in a language like Chinese. Written Chinese does not mark word boundaries. Instead, each Chinese character is written one after the other without spaces. Since each character usually represents a meaningful unit and since words can be composed of one or more characters, it is often difficult to know where words should be segmented. Proper word segmentation is necessary for many applications particularly including parsing and text-to-speech. The way a sentence is broken up into words influences its pronunciation. The identification of correct word boundaries is a very important part of this process.

Another example is in the domain of biological sequences. The central dogma of molecular biology states that the flow of genetic information is from DNA sequences to RNA sequences to protein sequences. Genes, which are parts of DNA sequences that store genetic information, are transcribed (that is, copied), to messenger RNA (mRNA). The mRNA sequence carries this information outside the cell nucleus into the cytoplasm. There this information is translated into proteins. However, this process is more complex than the mere copying of sequence letters. In eukaryotic organisms, protein-coding genes are often interrupted by intervening sequences, called introns, and these need to be recognized and removed from mRNA before it is translated in order to produce

functional proteins. The classification of sequence segments into correct categories is a fundamental part of this process.

Sequence classification is an important problem that arises in many real-world applications: text classification, speech recognition, intrusion detection, and protein-coding sequence prediction, among others. Given a sequence, the task of a sequence classifier is to assign a class label to that sequence. In this context, “sequence” means an ordered combination of letters drawn from a finite alphabet; for instance, a vocabulary of English words in the case of text classification, the four-letter alphabet of nucleotides in the case of coding-sequence classification. Class labels, however, are typically drawn from a finite set of mutually exclusive choices, i.e. parts-of-speech: noun, pronoun, etc.

In many domains, sequence data contain useful “signals,” or features, that enable the construction of classification algorithms. In handwriting recognition, the features may include horizontal and vertical profiles, internal holes, strokes, and other characteristics of handwritten characters. In speech recognition, the features may include phonemes, noise ratios, length of sounds, and more. In the domain of spam detection, examples of features are email headers (their presence and form), grammar characteristics, n-gram frequency counts, and more. In biological sequence-classification problems, gene- and protein-sequence features may be nucleotide or amino-acid blocks, their respective positions in the sequence, as well as many other possible combinations.

In all these cases, extracting and interpreting the most useful features is challenging, and hand construction of good features is the basis of almost all classification algorithms. Automatic methods usually use a “brute force” approach, in which sequence-classification models are provided with a vast number of features,

hoping that the important features will not be overlooked. The large number of features introduces a dimensionality problem having several disadvantages: enumerating all possible features is impractical; many features are irrelevant to the classification task and affect accuracy adversely; and knowledge discovery is complicated by the large number of parameters involved.

Feature-selection techniques are used to select a representative feature set from the available features for classification algorithms. A feature-selection technique may use the intrinsic properties of the dataset or the classification goal and the classification algorithm as a guide for heuristic searches to find a useful and informative set of features from a large collection of features. When the large collection of features is pruned by a feature-selection technique, its size is reduced, leaving useful features for the task at hand.

In this thesis, we develop a scalable method for automatic feature generation for sequences. The algorithm uses sequence components and domain knowledge to systematically construct features, explores the space of possible features, and identifies the most useful ones. This focused feature-generation algorithm (FGA) integrates feature construction and feature selection in a systematic way. We show that, coupled with an appropriate classification algorithm, FGA is very effective in the task of sequence classification [26,29].

To demonstrate our algorithm, we implement a novel splice-site prediction procedure for mRNA sequences. Splice-sites are the boundaries of introns in the primary mRNA (pre-mRNA) transcript, and splicing is the process that involves the excision of introns and the ligation of exons to form the mature mRNA, ready to be translated to

protein. This process is one of the essential cellular processes in eukaryotic organisms and, although it has been studied extensively, many unanswered questions still remain. A crucial one is: how are the splice sites accurately identified and correctly paired across the intron? It is currently believed that identification is accomplished, at least partially, through the conserved sequences at the sequence boundaries. However, these conserved sequences are short and not well defined, and are often hard to distinguish from the numerous, unutilized sequences throughout the genome.

Elucidating the complex details of the gene-splicing process is of significant importance for biology and medicine: it has been estimated that ~ 15% of human genetic diseases are caused by errors in splicing [34]. Understanding splicing is a major step towards understanding these diseases. Furthermore, improved understanding of splicing directly impacts computational gene finding [38]. This is in the form of better computational models and increased prediction accuracy. Today, computational gene finding is arguably the most important task in genomics. Large quantities of genomic sequences are generated daily by numerous gene-sequencing projects, and their accurate annotation by gene-finding algorithms presents a major challenge.

In bioinformatics, automatic sequence classification has many other applications ranging from the implementation of fast database searches to the identification of patterns for some specific physical traits.

In addition to its value in sequence classification, a carefully selected set of features has other notable benefits. For a biologist trying to identify the signals or patterns that contribute to splicing, for example, the features generated by FGA provide a good hypothesis set. Rather than trying to guess the critical sequence elements and to

validate them by expensive experimentation, the biologist can start with the high-scoring features from FGA, thereby significantly reducing the amount of experimentation needed. We compared FGA features with known signals in literature and were able to show that many high-scoring features of FGA did correspond to functional elements [27]. So that biologists might take advantage of the features discovered by FGA for splice-site prediction, we created SplicePort [28], a web-based tool for splice-site analysis. SplicePort allows the user to make splice-site predictions for submitted sequences. The user may also browse the rich catalog of features underlying the predictions. Selected feature sets may be searched, ranked, or displayed. The user may then browse clusters of features and their contributing elements, look for new and interesting signals, or validate previously observed signals.

1.1 Contributions of this thesis

This thesis presents, evaluates, and details an automatic feature-generation algorithm for sequence classification. Our contribution also extends to the field of bioinformatics, since we employ the feature-generation algorithm as a tool to study the gene-splicing problem. Specifically, our contributions are as follows:

Catalog of sequence features: We begin our study by categorizing the basic feature elements for sequences and we build a catalog of generic sequence feature types, along with the corresponding feature-construction methods for each of them. The feature-construction methods iteratively build composite features using the basic feature elements. For the problem of splice-site prediction, we generate a rich catalog of features capturing the compositional and the positional information of the splice-site sequence.

Feature-generation algorithm: The feature types are integrated into a scalable method for automatic feature generation: the feature-generation algorithm (FGA). The algorithm systematically constructs features, explores the space of possible features, and identifies the most useful ones. FGA integrates feature construction with feature-selection methods in order to identify a final set of informative features. An appropriate classifier learns the sequence-classification model, using the FGA identified set of features. FGA is a flexible, modular algorithm that can be easily adapted to any sequence-classification problem by identifying the most appropriate feature-selection method and most effective classifier.

Splice-site predictor: Using FGA, we identified a mix of features that, when used to build splice-site detection classifiers, achieved results that were significantly more accurate than those from existing, state-of-the-art splice-site-prediction algorithms.

SplicePort: We developed an interactive feature-browsing and visualization tool for splice-site analysis. This web-based tool allows the user to make splice-site predictions for submitted sequences based on our FGA analysis. The user can also browse the rich catalog of features underlying the predictions. Then, the user can view and explore subsets of splice-site-prediction features — either the features that account for the classification of a specific input sequence or the complete collection of features.

The web server is also intended to make the method generally applicable to different species without any major changes and with an equivalent performance. Feature-motif exploration enables researchers to rapidly explore the space of computationally identified signals and effectively pose hypotheses for experimental testing and validation. We explore the knowledge-discovery power of our algorithm for

the splice-site prediction problem by looking closely at the generated features, using SplicePort functions. The features detect biological signals, which may be important for the splicing process. The generated features for splice-site prediction include known functional elements and can be used to infer novel aspects of the splicing signal.

Structure features: We consider a different FGA role: can we extend the algorithm to model features that describe properties that are more complex? We employ some modifications to our initial features-construction methods, and we construct features that capture the three-dimensional structure of the pre-mRNA sequence near the splicing signals. These features capture specific structural contexts that indicate a significant influence of the secondary-structure properties upon gene-splicing. To explore the potential of the newly discovered features, we again use the SplicePort web server. Finally, we extend the splice-site model to include both the pre-mRNA sequence and structure characteristics. The new model significantly outperforms the sequence-based features model.

1.2 Organization of the thesis

Chapter 2 provides the necessary background for the topics covered in this thesis, gives an overview of the biological literature that stimulated our research, and we discuss the existing literature on computational splice-site prediction models. Chapter 3 describes the feature-generation algorithm detailed for sequence data, the different feature types describing sequence properties and their construction algorithms, feature-selection methods, and the Least Squares classifier. The latter part of the chapter details the experiments for splice-site prediction. Chapter 4 discusses the motivation for knowledge discovery and feature-space exploration to find biologically meaningful signals. The

chapter also describes the SplicePort web server and its rich functionality. Chapter 5 discusses the knowledge-discovery power of the feature-generation algorithm, illustrated with biologically relevant signals that we find, and their supporting literature. Chapter 6 revisits the splice-site problem and its definition, describing also the three-dimensional shape of a splicing event. The potential of the structural properties motivate our feature-generation algorithm for secondary-structure features. We describe new feature types and their construction methods, as well as the experiments that validate the secondary-structure importance of gene splicing. This chapter also summarizes and discusses the results obtained and outlines suggestions for further research.

Chapter 2: Splice-site prediction

This chapter describes the necessary background to understand the important process of gene splicing as a biological event. Splicing was discovered in 1977 [3,11]. This crucial discovery made clear that the gene was not a simple unit of heredity or function, but rather a series of exons, including the coding information for proteins and separated by long non-coding stretches called introns. Here, we give a simplified overview of the splicing mechanism and only briefly mention the complex proteins in the cell nucleus that regulate and facilitate this process. Next, we describe a set of existing computational methods for predicting splice sites. These sequence-based approaches are only a sample of the large body of literature on splicing, but provide the motivation for our research.

2.1 Genetic information

Deoxyribonucleic acid (DNA) is a nucleic acid molecule in the form of a twisted double strand that is the major component of chromosomes and carries genetic information. DNA, which is found in all living organisms except some viruses, is responsible for passing along hereditary characteristics from one generation to the next. Ribonucleic acid (RNA) is another nucleic acid found in all living cells that is essential for the manufacture of proteins according to the instructions carried by DNA. RNA has only one strand. The basic units capable of transmitting hereditary characteristics are called genes. A gene consists of a specific sequence of DNA found in a fixed position on a chromosome. The majority of genes code for proteins. Proteins are essential substances for the structure and function of all living cells and organisms.

The flow of genetic information, according to the central dogma of molecular biology, is from DNA to RNA to protein. The gene sequences of DNA serve as templates for the synthesis of messenger RNA (mRNA) molecules, in a process known as transcription. Messenger RNA carries this information outside the cell nucleus into the cytoplasm, where it is translated into proteins (Figure 2.1). In eukaryotic organisms, such as plants, and animals, protein-coding genes are often interrupted by intervening sequences, called introns, which must be removed from mRNA in order to produce functional proteins. The cellular process that involves the excision of introns from the primary mRNA transcript and the ligation of remaining exons into the mature mRNA is called splicing. The mature mRNA transcript, then, is transported outside the cell nucleus to ribosomes, where the information encoded in the nucleic acid sequence is translated to an amino acid sequence and converted into protein during the process of translation.

The DNA sequence is composed of four different nucleotides: Adenine, Cytosine, Guanine and Thymine (A, C, G, and T). The mRNA sequence also contains four nucleotides, with the exception that Thymine is substituted for Uracil (U). Amino acids (of which there are 20) are the building blocks of proteins. A string of three consecutive nucleotides (codon) codes for one amino acid. Several amino acids are coded by more than one codon. The protein code begins with the start codon (ATG) and ends with one of the three possible stop codons (TAA, TAG, or TGA). The coding regions of mRNA are usually preceded and succeeded by untranslated regions, which stabilize mRNA molecules and improve translation efficiency. These regions do not code for protein but serve as regulatory sequences.

2.2 Pre-mRNA splicing

The boundary locations between exon and intron regions are called splice sites. Splice sites are either acceptor sites, which mark the beginning of an exon, or donor sites, which mark the end of an exon, as shown in Figure 2.2. The entire coding region of a gene, as well as the untranslated regions of the mRNA (the "5' UTR" and "3' UTR") lie within the exons.

Splice-site signals (Figure 2.3) are short sequences of nucleotides preferred in the immediate splice-site neighborhood. Most introns start with the dinucleotide GT (GU in RNA sequence) and end with the dinucleotide AG (in the direction 5` to 3`). The occurrence of these specific dinucleotides, upstream and downstream, is not sufficient to signal the presence of an intron. Generally, the donor splice signal is conserved better than the acceptor splice signal, which is harder to recognize. Another distinct signal is the branch site, with consensus sequence YTRAY, where Y stands for pyrimidines (C or T) and R stands for purines (A or G). The nucleotide A is believed to be generally conserved, and found in all genes. Its location varies; typically, it is found 30 nucleotides upstream of the acceptor site, but it can also be found as close as 11 or as far as 100 nucleotides upstream. Another signal preceding the acceptor splice site is the pyrimidine-rich region.

In cells, the splicing process is usually catalyzed by a large protein complex, called a spliceosome, which consists of five small nuclear RNAs (U1, U2, U4, U5, and U6) and numerous other splicing factors. Splicing occurs in two consecutive chemical reactions. In the first reaction, the donor splice site at the 5’ exon/intron junction is cleaved and the intron 5’ end is ligated to the branch point. In the second reaction,

cleavage of the acceptor (3') splice site releases the intron as a lariat structure and 5' and 3' exons are joined together (Figure 2.4). Splice-site recognition and spliceosome assembly occur simultaneously: the 5' splice site is initially recognized through interaction with the U1 molecule [41]. In human and other similar organisms, this base-pairing interaction involves approximately nine nucleotides (nt), encompassing the last two or three exonic nucleotides and the first five or six nucleotides of the intron. Subsequently, the branch-point sequence base-pairs with U2 [4]. The other three snRNAs are then added to this complex through other base-pairing interactions. The complex then undergoes a series of structural rearrangements and is capable of catalyzing splicing reactions [53]. This summary is a simplified overview of this complex event.

Splicing of introns must be performed with single-nucleotide precision in order to produce functional proteins. This requires that the actual splice sites be accurately recognized and correctly paired across the intron. The recognition of splice sites is, at least partially, achieved by interaction between some spliceosomal snRNAs and short consensus sequences located at the 5' splice site and the branch point (an example for human introns is given in Figure 2.3). Conserved sequences are also found at the 3' splice site and in the form of a polypyrimidine tract (located immediately upstream from the 3' splice site), which mediate splicing through their interactions with splicing factors.

However, these consensus sequences are not uniquely associated with functional splice sites; there are numerous occurrences of these signals throughout the genome not utilized by the splicing machinery. This is illustrated in a study by Sun and Chasin [54], where positional weight matrices (described in the next section) were trained on 2400 instances of real human donor and acceptor sites to search for splice sites in the 42-kb

human hprt gene, which contains eight introns. This approach identified eight real donor sites along with over 100 pseudo donor sites with scores higher than that of lowest scoring real donor site. The results were even more discouraging for acceptor sites, since 683 pseudo sites were predicted. Not yet fully understood is how the precise specificity required to distinguish correct splice sites from similar “pseudo-sites” is achieved or how the correct donor/acceptor pairs are brought together.

2.3 Splice-site prediction approaches

The accurate location of splice sites is vital in gene finding. Gene finding is one of the first and most important steps in understanding the genome of a species once it has been sequenced. In eukaryotic organisms, especially complex organisms such as human beings, gene finding is challenging because of the splicing mechanism. Typically, a protein-coding human gene sequence can be divided into a dozen exons, each often less than two hundred nucleotides in length, some as short as ten or twenty. It may also include an exceptionally long exon, extending more than a thousand nucleotides. The design of a highly effective computational approach is complicated by the absence of a discernable pattern for sequence characteristics, such as the pre-mRNA sequence length, coding sequence length, and the number and length of exons and introns.

Relying on biological knowledge and results, researchers in computational biology approach this problem by modeling consensus sequences around splice sites and within introns. Various methods are used to model splicing signals, such as the simple consensus sequence model, which either looks for a specific sequence motif or allows some alternative nucleotides at certain positions in the motif; position-weight matrices, which represent the frequency of appearance of the A, C, G, and T nucleotides at each

position of the consensus sequence; and weight arrays, which exploit statistical dependences between adjacent nucleotides [8,19,47]. Weight matrices and weight arrays are used to score candidate sequence motives.

The weight matrix model (WMM) [51] computes the probabilities of nucleotides in each position in the splice-site sequence, assuming independence between positions. The weight array model (WAM) [62] extends WMM by taking into account the dependencies between adjacent nucleotides in the sequence. The maximal dependency decomposition (MDD) [8] is a decision-tree model that improves on previous models by capturing dependencies between non-adjacent, as well as adjacent nucleotides in the splice-site sequence.

These sequence models are usually not used in isolation; rather, they are integrated with content models that use coding statistics to distinguish between coding and non-coding regions. Integrated approaches can either be stand-alone splice-site predictors or gene finders that attempt to identify entire gene structures (splice sites in intron-containing genes and in the boundaries of coding regions). These methods yield better accuracy for splice-site recognition because they eliminate false positive splice sites that do not have the necessary shift in coding potential [7]. There are a number of methods used to combine signal detection with coding statistics for stand-alone splice-site prediction, including neural networks [25]; Bayesian networks [1,13]; rule-based expert systems [55]; and discriminant analysis [50].

GeneSplicer [45], proposed by Pertea et al., is a state-of-the-art computational tool for splice-site detection tool that employs a combination of MDD and Markov modeling techniques. GeneSplicer looks at splice sites which are boundaries for coding

exons and non-coding regions. GeneSplicer considers a splice site as a complex entity and is based on the following premise: since a coding-region splice site (by definition) is surrounded by a coding region and a non-coding region, a splice-site model should consider the coding difference between the two regions. GeneSplicer models the splice-site signal and the coding content in the upstream- and the downstream-sequence regions.

The GeneSplicer algorithm combines three different models for splice-site prediction. First, the statistical model of the immediate neighborhood of the site is, essentially, an MDD tree, modified so that a first order Markov chain, instead of a WMM, is built for each leaf of the decision tree. The other two models are second-order Markov chains trained on coding and non-coding sequences. They collected sequences of 80 nucleotides on either side of the true splice-sites, grouped them into coding and non-coding sets, and then used these data to build the Markov models. For exons and introns shorter than 80 nucleotides this procedure includes sequences from both coding and non-coding regions. But, since this only slightly changes the Markov probabilities, it is considered acceptable. Then, the final prediction for a given sequence is a combined score, the sum of the contributions of the three models. GeneSplicer is an accurate splice-site predictor that has successfully combined the signal statistical models (WAM and MDD to capture the consensus signal), with the content-sensor methods (Markov chains to capture coding/non-coding compositional differences).

To analyze a genomic sequence in order to recognize a target signal, such as the splice site, it is important to use all the information that can be extracted from the sequence. Specific candidate features can be generated and evaluated according to their relevance. The ability to select the relevant features has been the focus of intensive

research. Recently, feature-selection techniques have received increased attention for biological-data applications. The following is a non-comprehensive list. Liu and Wong [37] gave a good introduction for filtering methods in the prediction of translation-initiation sites. Degroves et al. [16] described a wrapper approach that used both SVMs and Naive Bayes to select relevant features for splice sites. Yeo et al. [59] used a model based on maximum entropy, in which only a small neighborhood around the splice site was considered. Zhang et al. [66] proposed a recursive-feature elimination approach using SVM.

Splice-site prediction has been the focus of other works, such as [2,17,61], that reported promising results when compared with GeneSplicer, but it is difficult for a biologist to interpret the features employed in those models. Especially, it is challenging to relate them to actual biological signals. SpliceMachine [17] is similar to our approach; because both methods employ sequence-based features. The SpliceMachine application performs a series of feature-subset selection steps to find the best combination for an accurate splice-site prediction model. It details an extensive search for the best set of features, which is different from the guided feature-generation algorithm discussed here.

2.4 The feature-generation approach

The next chapter describes a new approach to biological-sequence classification in general and a new method of splice-site prediction in particular. The feature-generation algorithm uses sequence properties to automatically construct useful features. The features have two components: the sequence alphabet and relative position. Feature-construction procedures produce complex features, including features containing elements that are not directly adjacent, and features that may be associated with a range

of relative positions in the sequence. When new features are constructed, feature-selection techniques are employed to assess the constructed features and identify those most promising. Then, in an iterative fashion, feature construction procedures are employed again. When building features, this algorithm follows the GeneSplicer lead and considers a long subsequence window for splice-site prediction. The larger neighborhood provides information for less-prominent but important signals that are not usually considered in gene-finding models. Then, a classification algorithm uses the identified features to predict splice sites.

Features constructed using sequence-domain knowledge are important for knowledge discovery. Given a set of search and browsing procedures, molecular biologists can explore collections of such computationally identified signals to discover new motifs and, possibly, to guide them in experimental testing and validation.

2.5 Figures of Chapter2

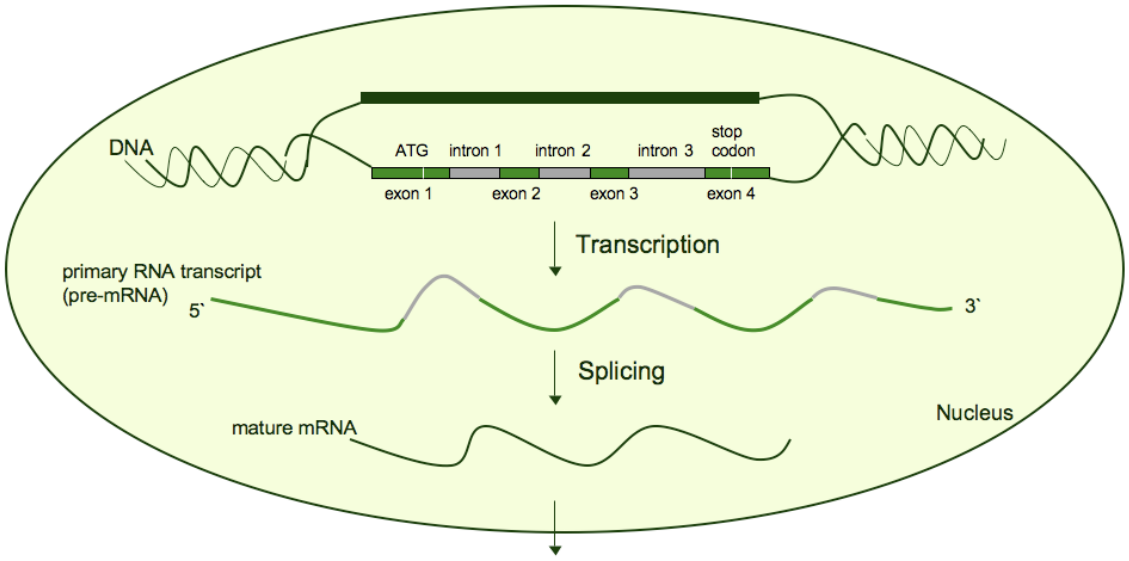


Figure 2.1. The flow of genetic information in a eukaryotic cell.

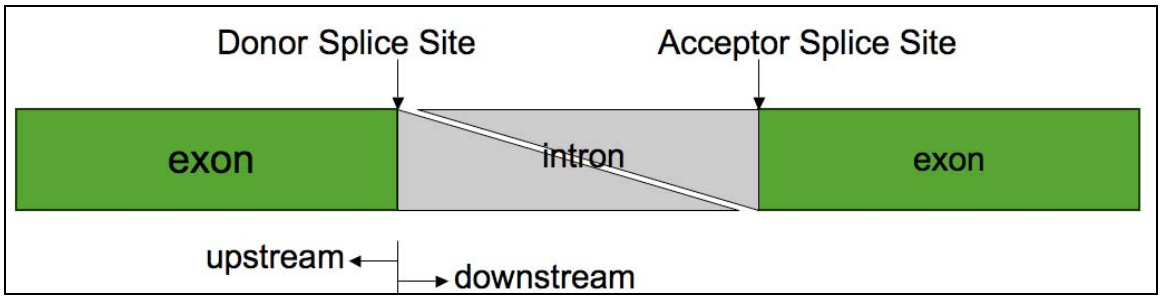


Figure 2.2. Depiction of a portion of a pre-mRNA transcript.

Splice sites mark the beginning (donor) and the end of an intron (acceptor). This figure shows an intron, which is removed from the RNA flanked by two exons. Real genes have a variable number of alternating exons and introns, and not all exons are protein-coding.

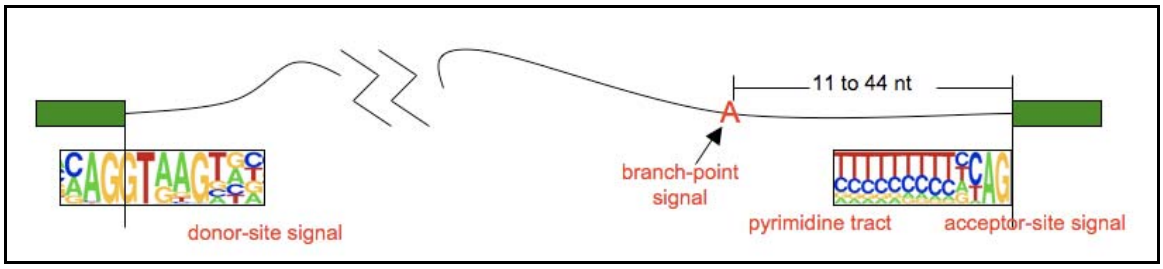


Figure 2.3. Signals involved in intron splicing in human genes.

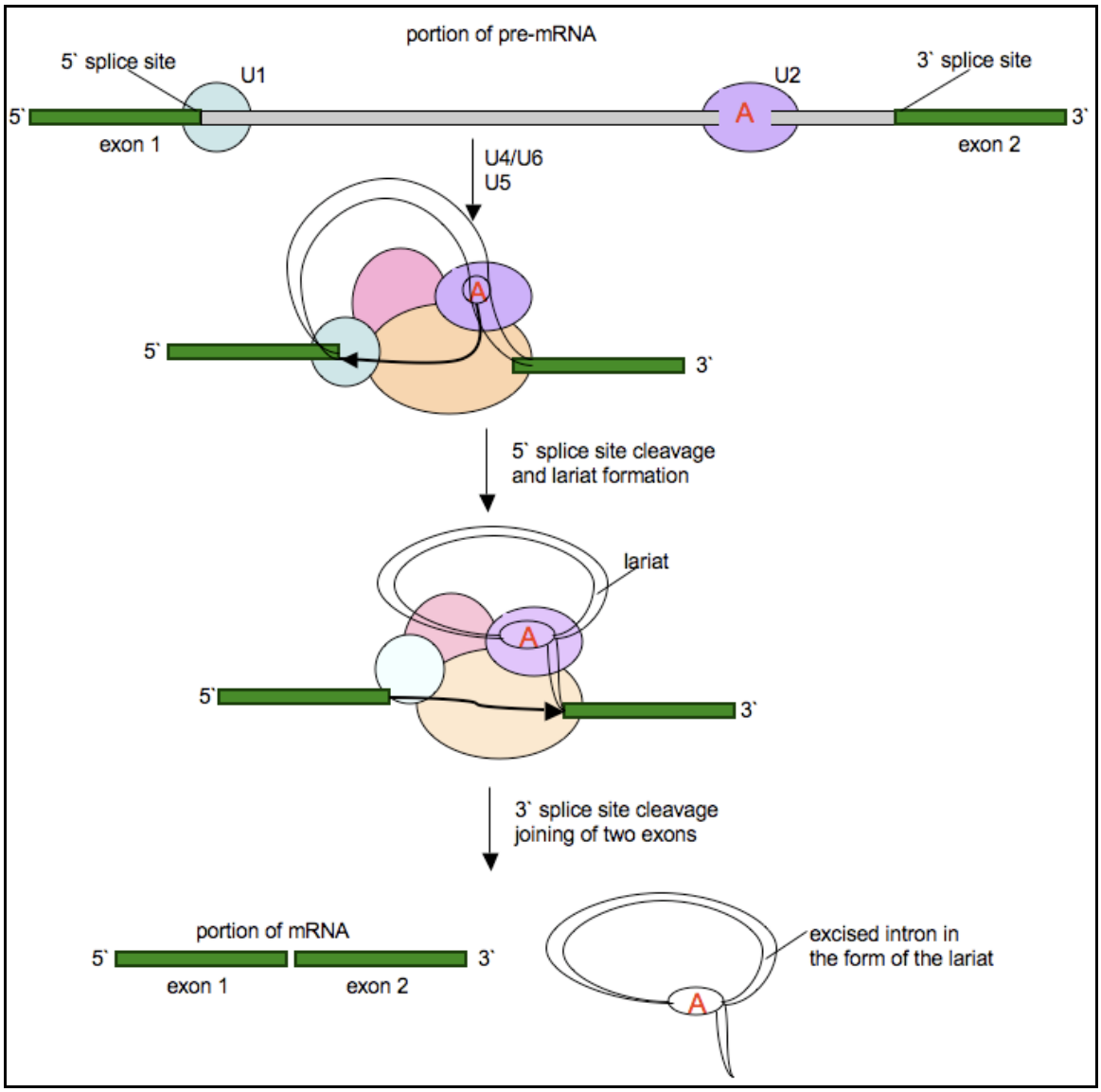


Figure 2.4. Splicing of pre-mRNA.

Chapter 3: Feature-Generation Algorithm

A good methodology for sequence-data analysis comprises these steps: (a) constructing candidate features from sequences, (b) selecting relevant features from the candidates and (c) integrating the final set of features in a system that recognizes specific properties in sequence data. Feature-generation algorithm, described in this chapter, is a process that integrates these three steps. The feature-generation process combines domain-specific feature-construction methods and off-the-shelf feature-selection methods. For generating candidate features, first, we present a catalogue of general sequence feature types, and then we describe their characteristics and the corresponding automatic construction methods. The starting points of the feature-construction methods are sequence alphabet (to construct words) and sequence-position information (to construct position-specific words). Then, a variety of operators, such as logical Boolean operators, are used to construct more complex features. These features have generic definitions and are suitable for various sequence data, not necessarily of biological origin. After describing feature construction, we next discuss different feature-selection methods and explain how they can be incorporated in the feature-generation algorithm to generate different feature types. Then we introduce a complete sequence classification framework based on the feature-generation algorithm. We discuss how to use such features to build reliable systems for sequence classification and present a thorough evaluation of the complete method using splice-site prediction as a benchmark problem.

3.1 Sequence data

The sequence data-classification problem is defined as follows. Given a set of categories C and a training set of sequences in each category, the goal is to learn a model so that for each previously unseen sequence, we can predict to which category it belongs. As an example, consider the protein-family classification problem. Given a set of protein families, find the family of a new protein. Moreover, consider the speech recognition problem. Given a set of utterances of a set of words, classify a new utterance to the right word.

Classification is an extensively researched topic in data mining and machine learning. Providing the assumption that training data has a fixed number of attributes, all of the existing classification methods may be used. In contrast, sequence data may possess no explicit features, as it is the case with DNA sequence data. In addition, sequences are of variable length with a special notion of order that may be important to capture.

To overcome these difficulties, the sequences that constitute the training set are usually restricted to a predefined length, and the sequence is represented as a vector of features, where each feature is a dimension and its coordinate value is a Boolean value, the aggregated count or some other computed score.

3.1.1 Sequence-data properties

A sequence is defined as a series of building blocks drawn from a pre-defined alphabet. For example, the building blocks may be the set of the twenty-six letters of the English alphabet. These form the words of the English language and words form an English-language document. In the case of biological sequences, the building blocks may

be the four nucleotides of the DNA sequence. Three-consecutive nucleotides form codons. These are the words that code for amino acids, the building blocks of proteins. A sequence of codons forms a protein-coding sequence, which in turn, translates into a protein.

In an English-language document, the identification of the correct meaning of any given sequence necessarily involves several knowledge sources, such as knowledge about the meaning of the words individually, knowledge of the grammatical structure of the sequence, knowledge about the context in which a particular word is occurring and common sense knowledge about the overall topic. Sequence composition is defined by the particular choice of words that describe the topic or topics of interest in the document. The relative positions of the words in the sequence, or their local context, i.e. the words "say" and "mean", and "eat" and "see" change the meanings of the sentences and therefore their topics of interest, in the following example.

As an example, consider the following excerpt from Lewis Carroll in Alice in Wonderland:

> "Then you should say what you mean," the March Hare went on.
>
> "I do," Alice hastily replied; " at least–at least I mean what I say— that's the same thing you know."
>
> "Not the same thing a bit!" said the Hatter. "You might just as well say that 'I see what I eat' is the same thing as 'I eat what I see'!"

DNA sequences, on the other hand, are examples of sequence data that possess no explicit words. Yet, a genomic sequence possesses biologically meaningful functional sites such as acceptor and donor splice sites that are associated with the primary structure

of genes. So, similar to the English language example, in Figure 3.1, we have two sequences with similar word composition, but different positioning of nucleotides. Both figures show two nucleotide-frequency plots, standing for acceptor and donor splice signals. The nucleotide-frequency plots depict the most common nucleotides found in those positions and they switch places in the figures. In both figures, they are linked by a string of 120 nucleotides. The relative positions of the splice signals change the meaning of the DNA sequences. In the figure on the left, we have the depiction of an exon of the average length in the human genome. In the figure on the right, we have the representation of a short intron of the human genome.

We say that, in a given sequence, the collection of words or building blocks that form the sequence defines its *compositional* information. In addition, the position of each present word in the given sequence, or the relative position of each present word with respect to other words in the sequence, defines its *positional* information. It is important that a catalogue of sequence of features captures both compositional and positional information. Accordingly, we use the sequence building blocks and their relative position in the sequence to define a series of feature types that capture these properties. For each feature type, we describe an incremental feature-construction procedure, which begins with an initial set of features and produces an expanded set of features.

3.2 Sequence feature types

We define a sequence S of length L as a string of L consecutive building blocks, $S = s_1 s_2 \ldots, s_L$, Each s_i denotes the sequence block in the i^{th} sequence position, for *i* equals *1,..,L*. For the DNA sequences, s_i is one of the characters {A, C, G, T}. We illustrate the feature types and the feature-construction procedures with examples on the DNA

sequence data, but the definitions and procedures apply to any sequence data defined over some fixed alphabet. We start with features that describe mainly the composition of sequences, and then we incorporate the position information.

3.2.1 Compositional features

A *compositional feature* is a feature that describes the sequence content. We distinguish several feature types:

General *k*-mers: A general *k*-mer is a string of k consecutive letters. For example, *ttta* is a general 4-mer defined over the DNA sequence alphabet, {A, C, G, T}. The general *k*-mers are useful for capturing information such as coding potential of sequences. For each general k-mer, we count the number of times the feature is present in the sequence. Consider the sequences S_A and S_D, shown in Figure 3.1. The value of the 4-mer *ttta* in sequence S_A is 2 and in sequence S_D is 3, because it occurs two times in S_A and three times in S_D. Given the four-letter alphabet for DNA sequences, the number of distinct *k*-mers is 4^k for each value of *k*. For all the sequences in our training set, we measure the general *k*-mer composition for *k* ranging from 2 to 6. For these values of *k*, there are 5,456 features.

Construction Method: Given an initial set of *k*-mer features, the construction method expands them to a set of *(k+1)*-mers by appending the letters of the alphabet to each *k*-mer feature. For example, suppose we begin with an initial set of 4-mers $F_{initial}$={*ttta*}. From that set, we construct the extended set of 5-mers $F_{constructed}$={*tttaa, tttac, tttag, tttat*}. In that manner, we incrementally construct level k+1 from level k. For the sequences S_A and S_D in Figure 3.1, the new constructed features will have these

values; S_A, (1,0,0,1), and S_D, (1,0,0,2), since *tttaa* occurs once in both sequences, *tttat* occurs once in S_A and twice in S_D but there are no occurrences of *tttac* and *tttag*.

Splice-site sequences of coding exons, which is the case in the sequences of our dataset, characteristically have a coding region and a non-coding region, as shown in Figure 2.2. For donor splice-site sequences bordering coding exons, the region of the sequence on the left of the splice-site position (upstream) is the coding region, shown in blue in Figure 3.1, and the region on the right of the splice-site position (downstream) is the non-coding region, shown in green in Figure 3.2. The opposite is true for acceptor splice sites of coding exons. The upstream region is part of the intron and the downstream region is part of the exon. These regions may exhibit distinct compositional properties. To capture these differences, we use *region-specific k-mers*.

Region-specific *k*-mers: A region-specific *k*-mer is a general *k*-mer found in a specified sequence interval, such as the upstream or downstream region. In this work, we consider both the upstream and the downstream regions. Other regions and interval specifications are also possible. For each upstream (downstream) *k*-mer, we count the number of times the feature is present in the upstream (downstream) neighborhood of the splice site. For example, in the sequences of Figure 3.2, the values of the upstream 4-mer *ttta* are: S_A, 1, and S_D, 1. Similar to general *k*-mers, for all the sequences in our training set, we measure the upstream and downstream *k*-mer composition for *k* values ranging from 2 to 6. This results in 10,912 potential features.

Construction Method: The construction procedure for upstream and downstream *k*-mer features is the same as the general *k*-mer method, with the addition of a region indicator. For the sequences S_A and S_D, the $F_{constructed}$={*tttaa, tttac, tttag, tttat*}^{upstream}

features will have these values; S_A, (1,0,0,1), and S_D, (1,0,0,1). For both sequences, *tttaa* and *tttat* occur once in the upstream region, and there are no occurrences of *tttac* or *tttag*. However, the $F_{constructed}$={*tttaa, tttac, tttag, tttat*}^{downstream} features will have these values; S_A, (0,0,0,0), and S_D, (0,0,0,1), since *tttat* occurs once in the downstream region of S_D, and there are no occurrences of the others.

3.2.2 Positional features

Position-specific nucleotides: A position-specific nucleotide, or a position-specific 1-mer, describes the occurrence of any particular nucleotide, {A, C, G, T}, at position *i* in the sequence. When many related sequences are aligned to the region of interest, the position-specific nucleotides capture the nucleotides preferred in certain positions. For example, when many splice-site sequences are aligned to the splice-site position, for both acceptor and donor sites, the frequency of observing each nucleotide in each sequence position is computed. Figure 3.1 shows the frequency plots of the acceptor and donor site signals. As shown in the figures, some nucleotides happen to be observed much more frequently in certain positions than others. The most frequent nucleotides for each position identify the consensus sequence, and all position-specific nucleotides, identify the position-specific matrix. Consensus sequences and position-specific matrices are used commonly in DNA sequence-classification analysis to describe various DNA sequence signals.

A position-specific nucleotide is a Boolean feature; for each feature we report if it is present in the sequence or not. As an example, assume that our feature set is $F_{initial} = \{a_1, c_1, ..., g_n, t_n\}$, where s_i denotes nucleotide *s* at the i^{th} sequence position. Our sequences have a length *n* of 162 nucleotides; therefore, our position-specific set of single

nucleotides contains 648 features. We use this initial feature set to construct complex position-specific features. For the sequences S_A and S_D of Figure 3.2, the feature set {a_1, c_1, g_1, t_1} will have the values; S_A, (0,0,0,1), and S_D, (0,0,0,1), since for both sequences the first nucleotide is T.

Position-specific *k*-mers: A position-specific *k*-mer is a string of *k*-characters that, starting at position *i* in the sequence, represents the substring appearing at positions *i, i+1,.., i+k-1*. These features are the most common features used for finding signals in the DNA sequence data. Position-specific 1-mers are a subset of position-specific *k*-mers; they are the simplest features of this type for the case *k=1*.

This feature type is useful for discovering species-specific functional signals, as well as evolutionary conserved functional signals. For the splice-site signal, these nucleotides are also of primary importance, as they may capture binding information. Position-specific *k*-mers capture the correlations between *k*-adjacent nucleotides. They are Booloean features. For each position-specific *k*-mer, we record the presence or absence of that feature in the neighborhood of the splice-site. As an example, for the sequences S_A and S_D in Figure 3.2, the feature $a_2a_3c_4a_5$ will have the values; S_A, 1, and S_D, 0, since the 4-mer *aaca* is present in positions 2, 3, 4, and 5, in sequence S_A, but not in S_D. This results in $(n-k+1)\times 4^k$ potential features for each value of *k* and sequence of length *n*.

Construction Method: This method starts with an initial set of position-specific *k*-mer features and extends them to a set of position-specific *(k+1)*-mers by appending the letters of the alphabet to each position-specific *k*-mer feature. As an example, consider an initial set of 2-mers, $F_{initial}$={a_3c_4, c_7c_8} where the subscript denotes the sequence

position. $F_{constructed}=\{a_3c_4a_5,\ a_3c_4c_5,\ a_3c_4g_5,\ a_3c_4t_5,\ c_7c_8a_9,\ c_7c_8c_9,\ c_7c_8g_9,\ c_7c_8t_9\}$ is the extended set of position-specific 3-mers. In this manner, we can incrementally construct level *k+1* from level *k*. For the sequence S_A feature set $F_{constructed}$ will have these values; (1,0,0,0,1,0,0,0), since $a_3c_4a_5$, and $c_7c_8a_9$ are the correct nucleotides in those positions. For the sequence S_D all these features will have the value zero, since none of these nucleotide combinations occur in those positions.

3.2.3 Composite positional features

Conjunctive position-specific features: A *k-nucleotide position-specific* feature is a set of *k* position-specific 1-mers combined with the logical operator AND. This feature type is useful for discovering interacting functional signals in the sequence. It captures the correlations between different nucleotides in non-consecutive positions in the sequence, and identifies the preferences of co-occurrence for not-necessarily-adjacent position-specific sets of *k*-nucleotides. A conjunctive position-specific feature is a Boolean feature. For each conjunctive positional feature, we record the presence or absence of that feature in the neighborhood of the splice site. Its dimensionality is inherently high. If the number of conjuncts is *k* in a given iteration, then we have a total of $\binom{n}{k}\times 4^k$ such features, for a sequence of length *n*.

Construction Method: We construct conjunctions of *(k+1)*-nucleotide position-specific features by starting with an initial conjunction of *k* position-specific features and, adding another conjunct feature in an unconstrained position. Let our position-specific nucleotides set be $F_{basic}=\{a_1,c_1,...,g_n,t_n\}$, where, a_1 denotes nucleotide *a* at the first sequence position and so on. If our initial set is $F_{initial}=\{a_1, g_2\}$, we can extend it to the

second level of position-specific base combinations $F_{constructed}=\{a_1 \wedge a_2, a_1 \wedge c_2,\ldots, g_2 \wedge t_n\}$, by forming a conjunction between every element of the initial set and every element of the basic set. Given an initial set of *k*-conjuncts, this construction method selects from the set of position-specific nucleotides to add another conjunct in an unconstrained position, thereby constructing the set of *(k+1)*-nucleotide conjuncts. A duplication check ensures that each feature in the $F_{constructed}$ set is unique. In this manner, we can incrementally construct higher levels.

Composite positional features: A *composite positional feature* consists of a conjunction of *n* nucleotides in *n* different positions co-occurring in the sequence. Composite positional features are a special case of conjunctive position-specific features. The difference is the initial feature set. Here, we start with a position-specific *k*-mer, and iteratively add other position-specific nucleotides in the nearby positions to form a composite positional feature. While the position-specific *k*-mers capture only the correlations among nearby positions, the composite positional features, similar to conjunctive position-specific features, are intended to capture the correlations between different nucleotides in non-consecutive positions in the sequence. An advantage of these features is their interpretability. Because we start with a given position-specific *k*-mer set, which serves a seed, these composite features are easier to interpret. The dimensionality of this kind of feature is still inherently high, but it is more restrictive than the conjunctive positional features. For a sequence of length *n*, if the initial feature set is the position-specific k_1-mer set, and the total number of conjuncts is n_1, $k_1 < n_1 << n$, we have a total of $(n-k_1+1)\times 4^{n_1} \times \binom{n-k_1}{n_1-k_1}$ such features.

Construction Method: Given the initial set of position-specific *k*-mers, this construction method selects from the set of position-specific nucleotides to add another conjunct in an unconstrained position. In this manner we construct the set of $(k+1)$-conjuncts. Now, if our initial set is $F_{initial} = \{a_1 g_2\}$, we can extend it to the composite positional features of three conjuncts: $F_{constructed} = \{a_1 g_2 \wedge a_3, a_1 g_2 \wedge c_3, ..., a_1 g_2 \wedge t_n\}$. Then, starting with the constructed composite feature set of three conjuncts, we can add another conjunct in an unconstrained position, to obtain a composite feature set of four conjuncts. Again, a duplication check ensures that each feature in the newly constructed set is unique. Incrementally, we can construct higher levels, in this manner.

Composite interval-specific features: A *composite interval-specific feature* is a composite positional feature that lies within a specified sequence region. The composite positional features, defined above, are obtained using position-specific nucleotides from the original sequence of length *n*. A specified sequence region is a subsequence within the original sequence. The difference between a composite interval-specific feature and a composite positional feature is the initial position-specific features set. For example, a composite upstream-region-specific feature is constructed using an initial feature set from the upstream-region position-specific k-mer features, and is expanded using the upstream-region position-specific 1-mer features. This definition can be extended to other sequence regions or "user-defined intervals".

Construction Method: The user initially identifies the interval of interest within the original sequence, for example, the branch-site interval, involving the positions [-40,-20] in the acceptor-site sequence. The position-specific nucleotides of this specified interval form $F_{basic} = \{a_{-40}, c_{-40}, ..., g_{-20}, t_{-20}\}$. Then, the construction method starts with an

initial set of composite features associated with the given interval, for example $F_{initial} = \{a_{-30} g_{-29}\}$ and, for each iteration, it selects an additive conjunct, in an unconstrained position, from the newly defined basic feature set. In this manner, we construct higher levels.

An interval-specific composite feature can have, up to, n_{intv} conjuncts, where n_{intv} is the specified-interval length. In this case, the constructed set is a subset of position-specific n_{intv}-mers.

Composite relative features: The positional features discussed so far define patterns of nucleotides in particular sequence positions. However, a biologist may also want to discover patterns of nucleotides in relative sequence positions. Therefore, we define this specific feature type. A composite relative feature is a conjunctive pattern of *k*-nucleotides that is not tied to a specific position in the sequence. These features consist of basic conjuncts that belong to a short sequence window of length n_1, and the start of the first conjunct may be anywhere in the given sequence of length *n*, where $n_1 << n$.

Construction Method: For each relative composite feature we record the number of times that feature is present in the neighborhood of the splice-site. As an example, consider the feature *ta*c*, or $t_i \wedge a_{i+1} \wedge c_{i+3}$, and the sequence S_A in Figure 3.1. The feature is constructed from two conjuncts in the window of length four, and it occurs six times in the given sequence of length 162. A relative composite feature set may have up to n_1 conjuncts. If all the conjuncts are used, the feature set becomes a subset of general n_1-mers.

3.3 Feature-Selection Analysis

Feature-selection methods reduce the size of the constructed feature set, keeping only the features useful for the task at hand. The problem of selecting useful features has been the focus of extensive research and many approaches have been proposed [5,30,32,58,60]. Generally these approaches are divided into three major categories: Filter approaches use the intrinsic properties of the dataset, such as feature-class entropy, to compute a feature-relevance score. Low-scoring features are removed, independent of the classifier algorithm. These approaches are usually very fast and are primarily used for high dimensional datasets. Wrapper approaches constitute the second class of feature-selection methods. They perform a heuristic search through all the subsets using the classification algorithm as a guide to find promising subsets of features. These approaches have the disadvantage of being computationally intensive, which limits wrapper approaches to datasets of low-dimensionality. In the third group, embedded approaches, the feature-selection method makes direct use of the parameters of the learned model to include or reject features. In the following we take a closer look at the first group.

3.3.1 Filter-selection methods

In our experiments, we considered different feature-selection methods to reduce the size of our constructed feature sets. We used several filter approaches, including information gain (IG), chi-square (CHI), mutual information (MI) and KL-distance (KL), to prune the feature-type sets during the generation stage of our method. We define these measures following Yang and Pedersen in [58].

IG: IG is frequently employed as a feature-goodness criterion in the field of machine learning. It measures the number of bits of information obtained for category prediction by knowing the presence or absence of a feature. If the number of categories in the given dataset is *m*, the categories are $c_1, \ldots, c_m$, and *P* denotes probability, then the information gain of feature *f* is defined to be:

$$IG(f) = -\sum_{i=1}^{m} P(c_i)\log P(c_i) + P(f)\sum_{i=1}^{m} P(c_i / f)\log P(c_i / f) + P(\bar{f})\sum_{i=1}^{m} P(c_i / \bar{f})\log P(c_i / \bar{f}) .$$

MI: MI is a criterion commonly used in statistical modeling of word associations in natural language documents. The MI between a feature *f* and the class c_i is defined to be:

$$MI(f, c_i) = \log P(f, c_i) / P(f)P(c_i) .$$

We combine the category specific scores to find average mutual information value:

$$MI_{avg}(f) = \sum_{i=1}^{m} P(c_i) MI(f, c_i) .$$

CHI: The CHI statistic measures the lack of independence between feature *f* and the category c_i. The contingency table of a feature f and class c_i produces the following numbers: N_{fc}, the number of data points that contain feature *f* and belong to class c_i; N_{fn}, the number of times *f* occurs without c_i; N_{nc}, the number of times c_i occurs without *f*, and N_{nn}, the number of times neither *f* nor c_i occurs. Assuming the size of dataset it *N*, the CHI measure is defined as:

$$CHI(f, c) = \frac{N \times \left(N_{fc}N_{nn} - N_{nc}N_{fn}\right)^2}{\left(N_{fc} + N_{nc}\right) \times \left(N_{fn} + N_{nn}\right) \times \left(N_{nc} + N_{nn}\right)} .$$

KL: The KL criterion measures the divergence between the distribution of features present in a training sequence and the categories to which that sequence may belong [48]. KL is defined as follows:

$$KL(f) = \sum_{i=1}^{m} P(c_i / f) \log(P(c_i / f) / P(c_i))$$

.

In the experiments discussed in the next section, we found that MI performed best for selecting compositional features, CHI for positional features, and IG for composite features.

3.3.2 Logistic selection scheme

As we described in the previous section, the filter-selection method assigns a score to every feature in the feature set based on the intrinsic properties of the dataset such as feature-class entropy. Recall also that we obtain the composite positional features by adding a new nucleotide from any position in the sequence to the initial feature set. The empirical test we performed on the data suggested improvements in performance by adding a score that penalizes the distance, such that the farther away the position of the newly added nucleotide to the original feature is, the lower the score of the newly constructed feature. We normalized the distance values to a standard normal distribution. Then, we applied a logistic scheme to assign these scores to each of the features in the constructed set of positional features. Finally, each feature was assigned a score according to the following formula:

$$F_{score} = IG(f) + \exp\left(dist^{-1}\right)$$

3.3.3 Recursive feature elimination

After we generate features for each feature type individually, we collect all the selected features into a combined set. At this stage we perform another feature-selection procedure, the recursive feature elimination. This procedure is more expensive than the filter-selection methods, and it involves the classification algorithm. Starting with the mixed set, we learn a prediction model using a max-margin classifier. A max-margin classifier, such as a linear support vector machine (SVM), produces a decision boundary to discriminate between two different categories. Each feature is assigned a weight during learning. These weights define the decision boundary and can be used for feature ranking. Features with zero weights, or weights very close to zero, are assumed to not contribute to the classification task [66], and are therefore eliminated. We used a regularized least-squares classifier [64] to learn the decision boundary and the individual feature weights. We recursively trained the classifier, learned a new model, and eliminated a fixed number of features after each iteration.

3.4 Feature-Generation Algorithm (FGA)

Our algorithm for feature generation uses domain knowledge and data properties to construct and select useful features for the prediction task. Starting with an initial feature set, FGA iteratively calls a feature-construction method to expand the current feature set, and a feature-selection method to reduce the feature set size to manageable levels. After a specified number of iterations, the algorithm produces an output feature set. Those features are then used by a classification algorithm to perform sequence classification.

Traditional feature-selection approaches consider a single brute-force selection over a large set of all features of all different types. In contrast we find that by categorizing the features into different feature types, it is possible to apply appropriate construction and selection methods suitable to the different types. Thus, we can extract relevant features from each feature-type set more efficiently than if a single selection method had been applied to the whole set. The type-oriented feature selection approach allows the use of different feature selection models for each type set; for example, for a feature set whose dimensionality is not too high, one may use a wrapper approach in the selection step, while for a large feature type set, one may use filter approaches. Furthermore, this allows features of different types to be generated in a parallel fashion.

To employ the information embedded in the selected features for sequence prediction, we use the following three-step algorithm:

- **Feature Generation:** The first stage generates feature sets for each feature type. For each defined feature type, we tightly couple the corresponding feature-construction step with a specified feature-selection step. We iterate through these steps to generate richer and more complex features. Each iteration, we eliminate features that are assigned a low selection score by the feature-selection method.
- **Feature Collection and Selection:** We collect features of different types and apply another selection step. The selection method we apply is recursive feature elimination. We recursively train the classifier and remove the low-scoring features. We produce a final set of features originating from different feature types and different selection procedures.

- **Classification:** The last stage of our algorithm builds a classifier over the final set of features. The regularized least-squares classifier, C-Modified Least Squares (CMLS), described by Zhang and Oles in [64], is a max-margin classification method similar to SVM. Compared to SVM, CMLS has a smoother penalty function, which allows calculation of gradients that provide faster convergence.

While feature generation remains a computationally intensive process, the organization of the generation process according to the different types allows us to search a much larger space efficiently. Moreover, this feature-generation approach has other advantages, such as the flexibility to adapt each individual generation process with respect to the feature type and the possibility to incorporate the module in a generic learning algorithm. To deal with the large number of features, we use CMLS, which is very efficient at handling problems of this size.

The feature generation stage is also very generic and offers the flexibility to accommodate several different scenarios. This component may operate in the coupled or uncoupled mode, as shown in Figure 3.3. When the component is in the uncoupled mode, see Figure 3.3, the feature-construction and selection steps are independent of each other. All the features constructed in the iteration step *i*, regardless of the scores assigned by the feature-selection method, are used in the next feature-construction step. This mode allows even the low-scoring features to expand in the next iteration. In our experiments, we allow this component to operate in the uncoupled mode during compositional-features generation.

When this component is in the coupled mode, see Figure 3.3, the quality of the features produced by the feature-construction method in the next iteration depends on the ability of the feature-selection method to detect the useful features in the current iteration. The features that score below the decided threshold are not allowed to expand in the next iteration. This mode of operation is useful when the dimensionality of the feature set is very high, as in our experiments with composite-positional features.

3.5 The Regularized Least-Squares Classification Algorithm

Now we take a closer look at the third stage of our method, i.e. classification and explain in detail how we perform this step. For this we take the set of features coming from the generation step and feed it to CMLS, a least-squares classifier algorithm. We found, when compared to AdaBoost, Support Vector Machines, Naïve Bayes and Maximum Entropy, this was the classifier that consistently gave the best performance. This linear classifier has a performance similar to linear support vector machines, but with a much faster convergence and therefore a shorter training time. Here we give a description summary of the regularized least-squares classifier as described by Zhang and Oles [64].

A two-class classification problem is to determine a label $y \in \{1,-1\}$ associated with a vector x of feature values. A useful method for solving this problem is by using linear discriminant functions, which consist of linear combinations of the feature input values. For a training set of labeled data $(x_1,y_1),...,(x_N,y_N)$, where N is the number of examples in the training set, we seek a weight vector ω and a threshold θ such that $\omega^T x < \theta$ if its label $y = -1$ and $\omega^T x \geq \theta$ if its label $y = 1$. Thus, the hyperplane $\omega^T x = \theta$ would approximately separate the training examples into the two classes that they belong.

Adjusting the equation so that we may take $\theta = 0$, the error rate for the linear classifier with weight vector ω is $\sum_{i=0}^{n} s\left(\omega^T x_i y_i\right)$, where s is the step function $s(z) = \begin{cases} 1 & if\ z \le 0 \\ 0 & if\ z > 0 \end{cases}$.

The least-squares fit algorithm finds the linear separator $\hat{\omega}$ that minimizes the error. In the regularized least-squares formulation, a regularization parameter λ is added in order to ensure that the problem will always have a solution:

$$\hat{\omega} = \arg\min_{\omega} \sum_{i=0}^{n} \left(\omega^T x_i y_i - 1\right)^2 + \lambda\omega^2$$

The solution is given by $\hat{\omega} = \left(\sum_{i=0}^{n} x_i x_i^T + \lambda n I\right)^{-1} \left(\sum_{i=0}^{n} x_i y_i\right)$, where I denotes the identity matrix. Because of λ, the ill-condition problem has a solution and the inverse of the matrix can be computed.

This formulation is very similar to the standard linear support vector machine, differing only in that SVM explicitly includes θ into the equation as follows:

$$\left(\hat{\omega}, \hat{\theta}\right) = \arg\min_{\omega,\theta} \sum_{i=0}^{n} g\left(y_i\left(\omega^T x_i - \theta\right)\right) + \lambda\omega^2, \text{ where } g(z) = \begin{cases} 1 - z & if\ z \le 1 \\ 0 & if\ z > 1 \end{cases}.$$

Zhang and Oles discuss that the non-smoothness of the loss function $g(z)$ introduces difficulties for direct numerical optimization. So the standard support vector machine formulation is solved as a quadratic programming problem or in its dual formulation. The authors observe that if the loss function were replaced by a smooth function, then it would be much easier to be solved directly in its primal formulation.

A slight modification to the equation replaces the function $g(z)$ with a smoother function, $h(z)$, to allow for an efficient application of the direct numerical optimization.

$$\hat{\omega} = \arg\min_{\omega} \sum_{i=0}^{n} h\left(\omega^T x_i y_i\right) + \lambda\omega^2, \text{ where } h(z) = \begin{cases} (1-z)^2 & if\ z \le 1 \\ 0 & if\ z > 1 \end{cases}.$$

Zhang and Oles, then, modify the function $h(z)$ further, by generalizing its desirable properties: the new function, $f(z)$, should be relatively smooth, with a continuous first-order derivative, and a non-negative piece-wise continuous second-order derivative. This formulation is convex, with a unique local minimum, which is also the global minimum. Numerical optimization methods, such as line-search methods, can be implemented then to find the optimal $\hat{\omega}$. These methods are guaranteed to converge, however, they may result in small step sizes, which slow down the convergence. To overcome this, the authors introduce a continuation parameter $c \in [0,1]$ and the new loss function is $f_c(z) = \begin{cases} (1-z)^2 & if\ z \le 1 \\ c(1-z)^2 & if\ z > 1 \end{cases}$.

So, for every step, a new c is chosen that $1 = c_1 \ge c_2 \ge ... \ge c_K = 0$, and the function $f(z)$ is modified accordingly. This C-modified least-squares algorithm (CMLS), is not required to converge, however it has been shown to accelerate the rate of convergence.

3.6 Evaluation Metrics

The performance of a class discrimination model is usually measured using the following basic measures: the number of true positives (TP), the number of false positives (FP), the number of true negatives (TN), and the number of false negatives (FN). After executing the prediction algorithm on a held-out test dataset, we calculate these numbers as follows: TP is the number of correct positive classifications returned by the prediction algorithm, FP is the number of data points classified as positive by the

system, which are not actual positives, TN is the number of data points correctly classified as negatives by the prediction algorithm, and FN is the number of actual-positive data points, which are not retrieved by the prediction algorithm and therefore, are wrongly classified as negatives.

Precision and recall, then, the standard performance measures of a classification method are defined: $precision = \frac{TP}{TP + FP} \times 100$, and $recall = \frac{TP}{TP + FN} \times 100$. Precision, also known as positive predictive value, is the measure of how much of information that the system returned is correct. Recall, on the other hand, also known as sensitivity, is a measure of how much relevant information the system has extracted. In this sense, precision and recall are antagonistic to one another, since a conservative system that strives for perfection in terms of precision will invariably lower its recall score.

We evaluate the performance of our model and we discuss our results based on these performance evaluation criteria: 11-point average precision, false positive rate, and Receiver Operating Characteristic (ROC) analysis, which we define next.

11-point average precision: The average precision of 11 recall points (11ptAvg Precision) [57] is a numerical measure that represents that average precision of the prediction system. It is calculated as follows: For any sensitivity ratio, we calculate the precision at the threshold, which achieves that ratio. Precision, measures the proportion of the sequences scoring above the threshold that are correctly retrieved, in our case, these are true splice sites. The 11ptAvg is the average of precisions estimated at these sensitivity values 0%, 10%, 20%, ..., 100%.

False-positive rate: Another performance measure commonly used for biological data is the false positive rate (FPr) defined as $FP_r = \frac{FP}{FP + TN}$, where FP and TN are the number of false positives and true negatives, respectively, defined above. FP_r can be computed for all recall values by varying the decision threshold of the classifier.

ROC analysis: We also draw the ROC curve, which is the graphical representation of the sensitivity (on the y-axis) versus false positive rate (on the x-axis). False positive rate is the value we wish to minimize, and the ROC graph shows the tradeoff between sensitivity and false positive rate.

Alternatively, we also present our results in the form of precision-recall curves. These curves plot the average precision at a given level of sensitivity (recall). Another metric that attempts to combine recall and precision into a single value is the break-even point. The break-even point is the value where precision equals recall. We also present results using this measure. In all our experiments, the results reported use three-fold cross-validation.

3.7 Experiments and Discussion

3.7.1 Data Description

The dataset used for feature generation is a collection of 4,000 human RefSeq pre-mRNA sequences, generously collected and provided by Alexander Souvorov (personal communication). All the splice sites in these pre-mRNA sequences contain the consensus di-nucleotides AG for acceptors and GT for donors. Following the GeneSplicer format, we marked the splice sites and formed subsequences consisting of 80 nucleotides upstream and 80 nucleotides downstream from the sites. We constructed negative

examples for the acceptor or donor datasets by choosing random AG-pair or GT-pair locations that were not annotated splice sites and selecting subsequences as we did for the true sites. From the true splice-site sequences we excluded the sequences containing unknown nucleotides, and, similar to GeneSplicer, we counted only the splice sites bordering coding exons. We only checked that the site itself lie in a coding region, but did not put any restriction on how long the coding region should be. Thus, the acceptor site data contains 20,996 positive instances and 200,000 negative instances. The donor site data contains 20,761 true positive instances and 200,000 negative instances. This data contains more acceptor sites than donor sites. This is due to the fact that more donor-site sequences contained unknown nucleotides in the region considered.

For further evaluation, we tested the classification model of the final set of features on the B2hum dataset, provided by the GeneSplicer team. This dataset contains 1115 human pre-mRNA sequences. There is no overlap between the set of these sequences and those the FGA algorithm is trained on.

Next, we discuss the prediction of acceptor and donor splice sites using the feature-generation algorithm. Let us remind that acceptor splice-site prediction is considered a harder problem than donor splice-site prediction, which is characterized by a better-conserved sequence structure.

3.7.2 Feature generation

A primary step in the construction of solid classification algorithms is the collection of features that distinguish between the two classes of interest. In a divide-and-conquer fashion, we examine each feature type separately and produce a brief evaluation of the effectiveness of the different feature types, when used in isolation.

Compositional features and region-specific compositional features: *K*-mer composition plays an important role in distinguishing sites and functional regions. In this analysis, we aim to identify those *k*-mer features that can help recognize the splice sites. We start with sets of all *k*-mers for each value of k from 2 to 6 and examine each *k*-mer feature set independently. Because the number of features is not very large (we have 16, 64, 256 and 1024 features for k values from 2 to 6) we use the FGA uncoupled mode. In this process, we allow all the constructed features to expand in the next level. After each construction step, we applied each of the feature selection methods listed in Section 3.3.1, to give a score to every feature. We ranked the features according to their score in decreasing order and selected the top scoring half. For each selected feature set, we used the CMLS classification algorithm to measure the splice-site prediction performance. We discovered that, when we used the MI selection method, the splice-site prediction performance for each selected feature set was as good as the whole *k*-mer feature set for each value of *k*, for both general *k*-mer and region-specific *k*-mer features.

Figure 3.4 shows the process of feature generation for general and region-specific feature sets for donor and acceptor dataset. We show the accuracy results for each general *k*-mer and region-specific *k*-mer feature sets after each iteration. In these experiments, after ranking the features according to each feature selection score, we selected the top 50% for each value of *k*. These results are for the MI selection method, as it worked best for compositional features. The results show that *k*-mer features carry more information when they are associated with a specific region (upstream or downstream) and this is shown by the significant increase in their 11ptAvg precisions.

Positional features: Next, we examine each position-specific *k*-mer feature set. *K*-mer compositional features adjacent to a particular site position may be used to discriminate such a site. In this analysis, we explore *k*-values from 1 to 6. Similar to compositional features, we use the FGA in the uncoupled mode and we measure the performance for splice-site prediction, when we use the complete set of position-specific *k*-mers and when we select the top scoring 1000 features, for each value of *k*.

The prediction results for this feature type are shown in Table 3.1 for acceptor and donor splice-site prediction. After each generation step, we observe a gradual increase in performance until level 3; then, the performance gradually drops. This can be explained with the exponential increase in the number of features after each level; i.e. the feature set of position-specific 6-mers contains more than 600,000 features. We believe that, for k values 4, 5 and 6, we are experiencing a form of overfitting, because the number of features we are constructing is very large. In this case, we need a larger number of sequences in order to be able to distinguish between the two classes.

In Table 3.1 we also list 11ptAvg precision results for the position-specific *k*-mer feature sets on acceptor and donor data when we use the IG, MI, CHI, and KL feature-selection methods to select the best 1000 scoring features. The IG and CHI feature-selection methods have a similar behavior. Our paired-t tests for statistical significance on the difference between their results reveal that the differences in these values are not statistically significant. The results on position-specific 6-mer features on both datasets and position-specific 4-mer features for the acceptor data were statistically significant. The KL distance shows good performance initially, but does not work well for more aggressive feature selection. This is most relevant for the set of position-specific 6-mers,

where we have the largest reduction in feature set size. The MI method seems unreliable for the set of position-specific 3-mers for the donor data, but works well for the other cases. We choose CHI to work with this feature type, but IG would also be a good choice.

Conjunctive positional features: Finally, we examined conjunctive positional features. Small groups of nucleotides adjacent to particular site positions, not necessarily adjacent to each other, may show a tendency to co-occur; therefore, they may be used to discriminate the site. These feature sets are extremely large; for example, for just three conjuncts there are 40 million unique combinations. For this reason, and because of our experience with position-specific *k*-mer features, in these experiments, we use the FGA in the coupled mode. We explored sets of 2 to 6 conjuncts, denoted as P2, P3, P4, P5, P6. At each level, we used the IG selection method to select the top scoring 1000 features. We repeated the generation using this selected set to produce the next level of features.

Figure 3.5 depicts the performances of the conjunctive feature sets for acceptor and donor data. For comparison, we introduce a baseline method, which is the average of 10 trials of randomly picking 1000 conjunctive features from each level. We can see from the graphs in Figure 3.5 that feature generation algorithm is picking up informative features that help distinguish the true splice-site locations. The 11ptAvg precision of these feature sets gradually drops as we generate more complex features. This happens because the feature set that is explored grows exponentially with each addition of another conjunct. The difference in precision values however, between FGA and the baseline method is highly significant on every value of k (alpha=0.005). Moreover, the generated features of this type can capture important functional biological signals.

3.7.3 Prediction results for individual feature types

In the previous section, for each feature type, we produce a final set of features consisting of features from each construction level. For example, the final set of general k-mer features consists of general 2-, 3-, 4-, 5-, and 6-mers. Here, we compare these collections of different levels of the feature sets of different types. The results are summarized in Figure 3.6.

Compositional features and region-specific compositional features: For the compositional feature sets, during each iteration of the FGA algorithm, we used the MI selection method to reduce the number of features in half. Therefore, after collecting all the selected *k*-mer features, we have a total of 2728 features for *k* values ranging from 2 to 6 for general, upstream or downstream *k*-mers. In order to reduce these numbers further, we used the recursive feature elimination. We eliminated 100 features at a time and stopped when the cross-validated 11ptAVG value started to drop. The number of features and splice-site prediction results are as follows.

The first three bars in Figure 3.6, acceptor, show the results for the best *k*-mer features for k ranging from 2 to 6 on acceptor data. The general *k*-mer feature set contains 700 features and 11ptAvg precision is 39.84%. The upstream and downstream *k*-mer feature sets sizes are 1,500 features and 1,800 features, and their results are respectively 58.77% and 52.01%. Similarly, in Figure 3.6, donor, the first three bars summarize the results for the general and region-specific *k*-mer features on donor data. The general *k*-mer feature set contains 1000 features and its 11ptAvg precision is 47.82%. The upstream and downstream *k*-mer feature sets size is 1200 features each, and their results are, respectively, 62.52% and 60.65%.

Position-specific *k*-mers: The fourth bar shows the results for position specific 1-mers. The respective precision results are 80.27% for acceptor data and 82.11% for donor data. The next bar in Figure 3.6, acceptor, shows 5000 position-specific *k*-mer features selected using the CHI selection method. The 11ptAvg precision of this set is 85.94%. The result for 5000 position-specific *k*-mer features on donor data is 86.67% represented with the fifth bar in Figure 3.6, donor.

Composite positional features: The sixth bars on both graphs in Figure 3.6 show the results for composite positional features. For acceptor data we have a collection of 3000 composite positional features for k ranging from 2 to 6 selected using IG. The 11ptAvg precision that this collection set gives is 82.67%. The collection of composite positional features for donor data results in an 11ptAvg precision of 83.95%. These results clearly show that using complex position-specific features is beneficial. Interestingly, these features typically are not considered by existing splice-site prediction algorithms.

Figure 3.6 also shows the performance of GeneSplicer on the same datasets as the last bar in the graph. We see that even in isolation, our positional features and our composite positional features perform better than GeneSplicer. These results are also statistically significant.

3.7.4 Splice-site prediction with FGA features

Once we collected all the features that we presented in Figure 3.6, general *k*-mers, upstream/downstream *k*-mers, position-specific *k*-mers and composite position-specific features, we ran the CMLS classification algorithm for both acceptor and donor. We achieved 11ptAvg precision performances of 92.08% and 89.70% in the acceptor and

donor datasets, respectively. These improvements are highly statistically significant, (alpha=0.005 for both acceptor and donor).

The improvement is dramatic over one of the leading programs in splice-site prediction, GeneSplicer, which yields 11ptAvg precisions of 81.89% and 80.10% on the same datasets. The precision results of FGA-generated features at all individual recall points, shown in Figure 3.7, are consistently higher than those of GeneSplicer for both acceptor and donor site prediction. The break-even points for acceptor splice-site prediction for FGA and GeneSplicer are 67.8% and 54.9%, respectively. Donor splice-site prediction produced break-even values of 66.7% and 58.7%, respectively for FGA and GeneSplicer.

In Figure 3.8 we explore feature-selection options that are more aggressive, using the more expensive recursive feature-elimination method in order to select a smaller working feature set. Recursive feature elimination shows that the generated features using this algorithm are very robust. For donor splice-site prediction, even the feature set of size 500 yields an 11ptAvg precision of 89.66%. This is an improvement of 9.56% over GeneSplicer on the same dataset. For acceptor splice-site prediction, even the feature set of size 1000 yields an 11ptAvg precision of 91.01%. This is an improvement of 9.12% over GeneSplicer on the same dataset.

Next, for further evaluation, we tested both algorithms on the B2hum dataset provided by the GeneSplicer team, which contains 1115 human pre-mRNA sequences. The FGA final feature sets for acceptor and donor splice-site prediction contained 3000 and 1500 features, respectively. In Figure 3.9 we present the false positive rates for a range of recall values from 5% to 95%. Figure 3.9 shows actually ROC curves with the

false positive rate shown on the y-axis. (An ROC analysis describing FGA splice-site prediction in comparison with GeneSplicer and MaxEnt, is shown in Figure 3.10, for both acceptor and donor sites.) If we compare the AUC values for FGA and GeneSplicer, we get the following results. In the task of acceptor splice-site prediction, FGA algorithm and GeneSplicer score 99.37% and 98.71%, respectively. In the task of donor splice-site prediction, the AUC scores are 99.25% and 98.58% for FGA and GeneSplicer, respectively. The feature-generation algorithm, with its rich set of features, consistently performed better than GeneSplicer in the B2hum dataset as well (B2hum is the dataset the latter algorithm is trained on). FGA false positive rates, as depicted in Figure 3.9, are favorably lower at all recall values. At a 95% sensitivity rate the false positive rate decreased from 6.2% to 2.5% for acceptor and from 6.7% to 3.3% for donor splice-site prediction. This significant reduction in false positive predictions can have a great impact when splice-site prediction is incorporated into a gene-finding program.

It should also be noted that there is no significant difference in the running time of FGA compared to GeneSplicer. Once the final set of features is determined, FGA performs a linear search (in terms of sequence length) along the given sequence to find high scoring sites.

3.7.5 Splice-site prediction with other classifiers

Now we refer to the note we mentioned in Section 3.5 that CMLS, the least-squares classifier algorithm, gave the best performance when compared to AdaBoost, Support Vector Machines and Naïve Bayes. The SVM implementation available to us, at the time, was several times slower. As an example, it took three days to train SVM, but only four hours to train CMLS at the time we were doing these experiments. Today there

exist much faster SVM implementations compared to the time we started working on these experiments. In this respect, when deciding which classification algorithm to choose, the training time criterion is not the main constraint. However, as shown in Figure 3.11, the linear classifier has a performance almost identical to linear support vector machines. In addition, the individual numbers of these precision-recall curves for all the classification algorithms are detailed in Table 3.2. In this table, it is also shown that the classification performance of CMLS is better than the others (precision values are consistently higher than those of the other algorithms), even when compared to SVM performance. This difference is not statistically significant when we compare CMLS and SVM, but it is significant when compared to AdaBoost and Naïve Bayes.

3.8 Summary

We have presented a general feature generation framework that integrates feature construction and feature selection in a flexible manner. We showed how this method can be used to build accurate sequence classifiers. We presented experimental results for the problem of splice-site prediction. Using the feature generation approach, we were able to search over an extremely large space of feature sets effectively, and we were able to identify the most useful set of features of each type. By using this mix of feature types, and searching over their combinations, we were able to build classifiers that achieved accuracy improvements of 10.6% and 9.5% over an existing state-of-the-art splice-site prediction algorithm, GeneSplicer. The specificity values were consistently higher for all sensitivity thresholds and the false positive rate decreased favorably. We have also shown that some of these features describe biologically significant functional elements. They are freely available to all interested researchers, and can be viewed at www.spliceport.org or

http://www.cs.umd.edu/projects/SplicePort/. We describe these in the next chapter. This algorithm, with its systematic feature generation basis, can be applied to more complex feature types and other sequence-prediction tasks, such as translation start-site prediction, protein sequence-classification problems. Moreover, it can easily be extended to genomic data of other organisms.

3.9 Tables of Chapter 3

Table 3.1. Feature generation comparison for position-specific *k*-mer features for *k* from 1 to 6 for acceptor and donor splice-site prediction

We give the 11ptAvg precision for each set when all the features are used and for each selected set with different selection methods.

Acceptor

Pspec-Kmer	11ptAvg (Acc)	IG-1,000	MI-1,000	CHI-1,000	KL-1,000
1	79.85				
2	85.96	84.91	76.49	84.68	84.84
3	86.54	82.43	74.36	82.46	79.54
4	84.92	73.94	72.59	75.96	70.09
5	80.60	72.59	71.94	72.65	60.94
6	68.64	58.84	58.58	59.31	30.27

Donor

Pspec-Kmer	11ptAvg (Don)	IG-1,000	MI-1,000	CHI-1,000	KL-1,000
1	82.11				
2	86.47	85.61	82.75	85.02	85.20
3	87.46	84.58	65.42	84.45	84.06
4	87.31	80.80	79.15	80.77	77.18
5	86.31	80.34	80.93	80.48	77.77
6	84.93	68.94	70.16	70.35	47.21

Table 3.2. Comparison of precision-recall values for CMLS, SVM, AdaBoost and Naïve Bayes classifiers

We give the 11ptAvg precision for each.

Acceptor

Recall points	CMLS	SVM	AdaBoost	Naïve Bayes
0	1	1	1	1
0.1	0.9969	0.9966	0.9880	0.8768
0.2	0.9955	0.9944	0.9836	0.8701
0.3	0.9924	0.9932	0.9778	0.8611
0.4	0.9910	0.9908	0.9759	0.8516
0.5	0.9879	0.9878	0.9688	0.8389
0.6	0.9819	0.9808	0.9528	0.8176
0.7	0.9750	0.9711	0.9372	0.7855
0.8	0.9607	0.9554	0.9071	0.7408
0.9	0.9261	0.9205	0.8449	0.6580
1	0.3213	0.2867	0.1356	0.1448
11ptAVG	**0.9208**	**0.9161**	**0.8793**	**0.7677**

Donor

Recall points	CMLS	SVM	AdaBoost	Naïve Bayes
0	1	1	1	1
0.1	0.9944	0.9939	0.9765	0.9370
0.2	0.9923	0.9928	0.9740	0.9304
0.3	0.9900	0.9883	0.9696	0.9231
0.4	0.9852	0.9844	0.9676	0.9008
0.5	0.9798	0.9802	0.9625	0.8770
0.6	0.9746	0.9728	0.9525	0.8591
0.7	0.9625	0.9598	0.9349	0.8329
0.8	0.9445	0.9399	0.9117	0.7979
0.9	0.9118	0.9072	0.8475	0.7285
1	0.1926	0.1903	0.1354	0.1467
11ptAVG	**0.9025**	**0.9009**	**0.8756**	**0.8122**

3.10 Figures of Chapter 3

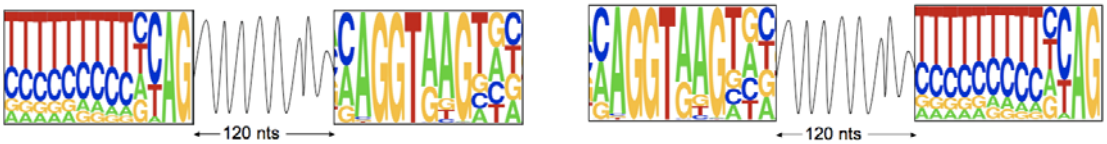


Figure 3.1. A schematic representation of a DNA sequence composition.

The figure on the left shows an acceptor-splice signal followed by a 120-nucleotide stretch, followed by the donor-splice signal. The splice signals are reversed in figure on the right.

Acceptor sequence (S_A):

```
TAACATCCATATAAAGCTATCTATATATAGCTAT   34
CTATGTCTATATAGCTATTTTTTTTAACTTCCTT   68
TATTTTCCTTACAGGGTTTCAGACAAAATCAAAA  102
AGAAGGAAGGTGCTCACATTCCTTAAATTAAGGA  136
GTAAGTCTGCCAGCATTATGAAAGTG
```

Donor sequence (S_D):

```
TTTAACTTCCTTTATTTTCCTTACAGGGTTTCAG   34
ACAAAATCAAAAAGAAGGAAGGTGCTCACATTCC   68
TTAAATTAAGGAGTAAGTCTGCCAGCATTATGAA  102
AGTGAATCTTACTTTTGTAAAACTTTATGGTTTG  136
TGGAAAACAAATGTTTTTGAACATTT
```

Figure 3.2**.** Sequence examples for acceptor (S_A) and donor (S_D).

The sequences consist of 162 letters each from the nucleotide alphabet {A, C, G, T}. The middle letters are AG for acceptor and GT for donor. The upstream region of the sequence is composed of the 80 nucleotides, shown in blue, and the downstream region consists of 80 nucleotides, shown in green.

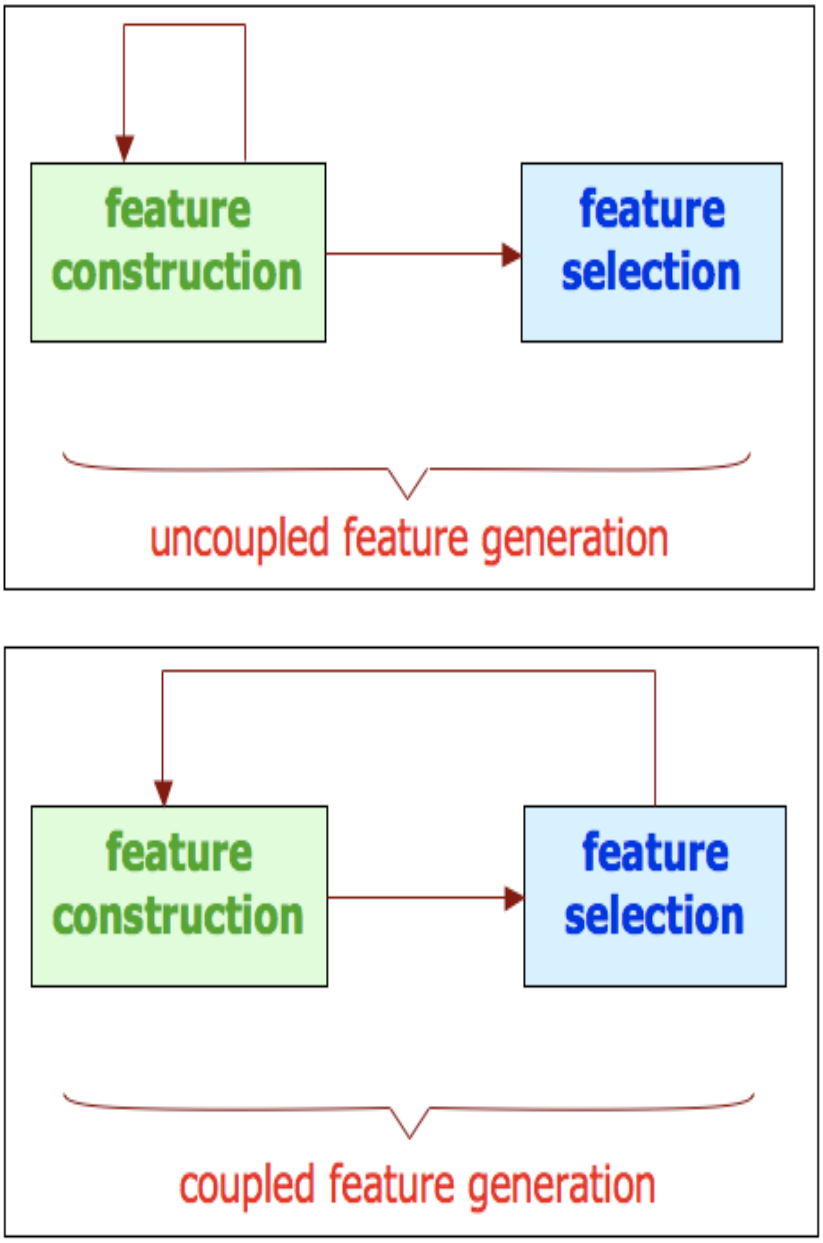
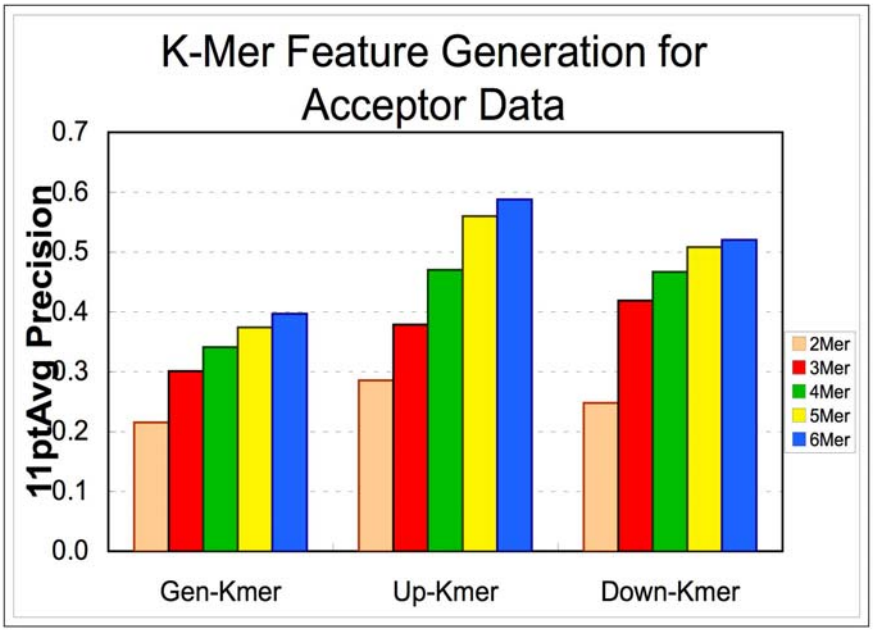


Figure 3.3. Feature generation component operating in uncoupled and coupled mode.

When feature generation operates in the coupled mode, features scoring below the decided threshold, after the feature selection step, are not allowed to expand in the next iteration.



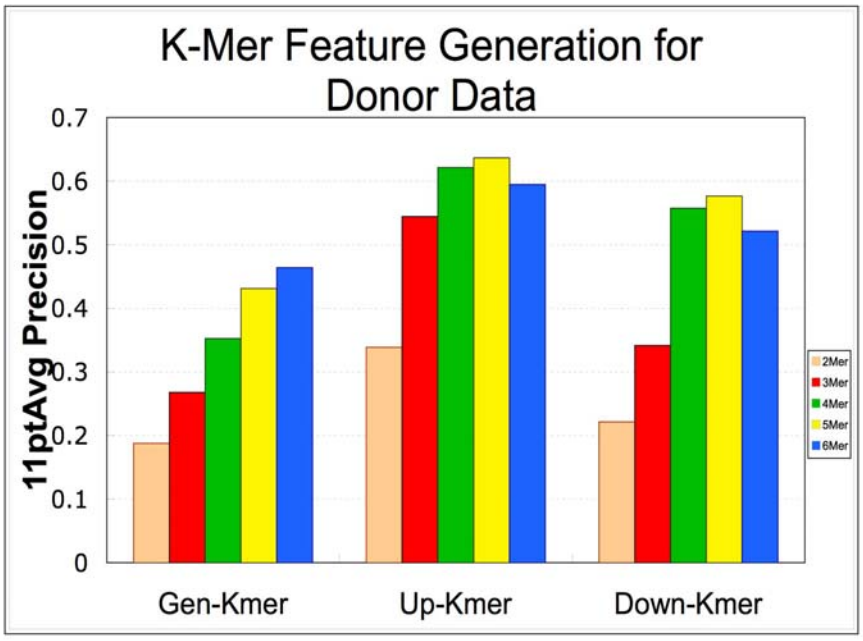
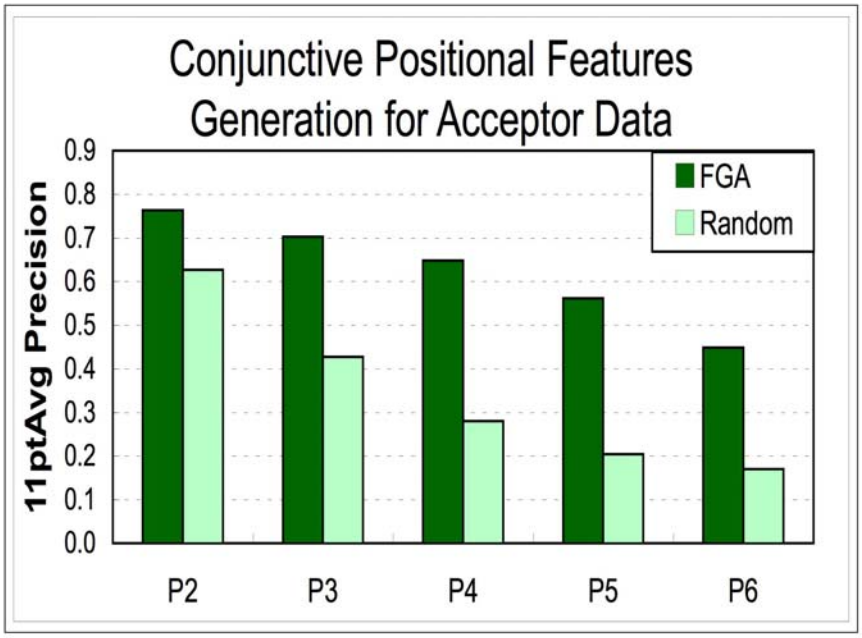


Figure 3.4. Feature generation comparison for performances of different feature type sets, general *k*-mers, upstream *k*-mers, and downstream *k*-mers, shown for different values of k for acceptor splice-site prediction and donor splice-site prediction.



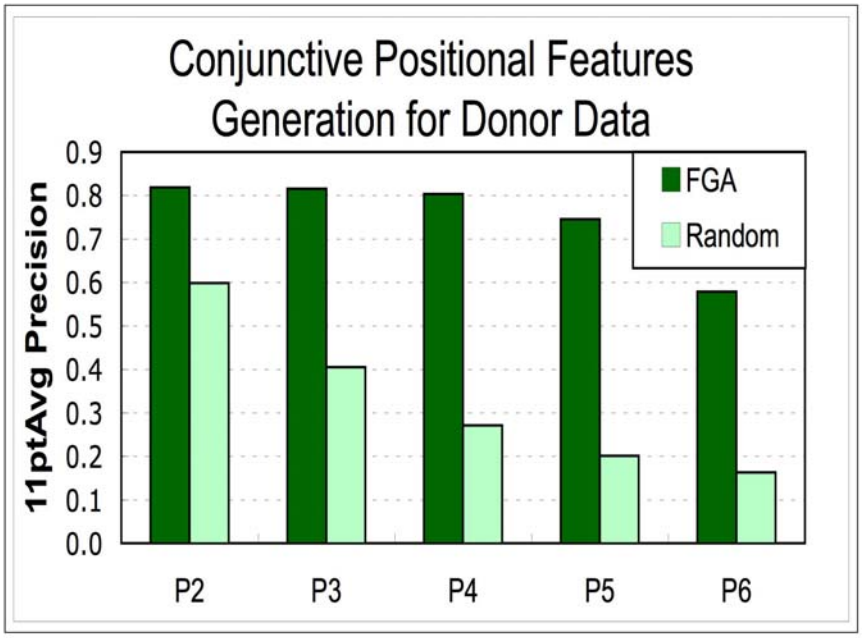
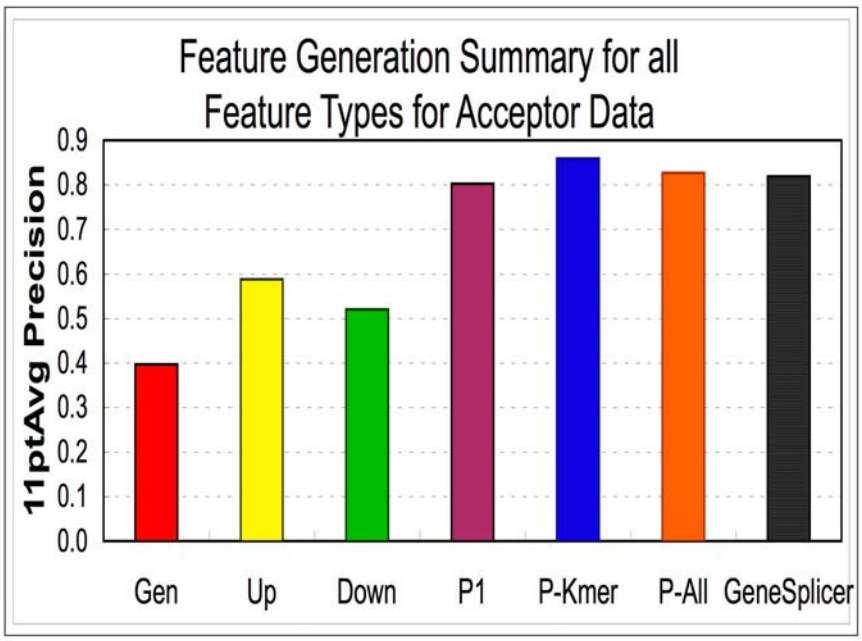


Figure 3.5. 11ptAvg results for the position specific feature sets generated with FGA algorithm vs. randomly generated features for acceptor and donor splice-site data.



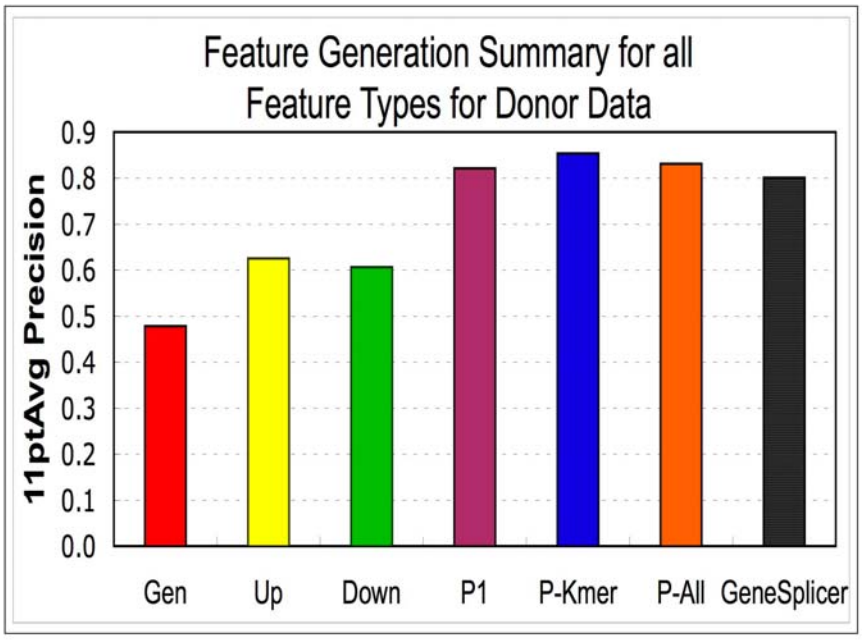
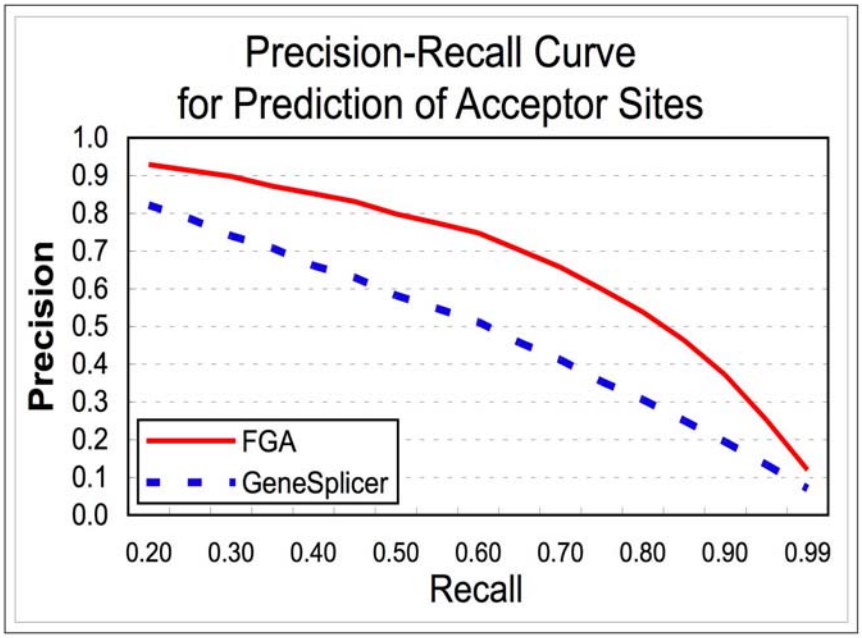


Figure 3.6. Performance results of the FGA method for different feature types as well as the GeneSplicer program in acceptor splice data and donor splice data.

The depicted feature sets are as follows: Gen - selected general *k*-mers, Up - selected upstream *k*-mers, Down - selected downstream *k*-mers, P1 - position-specific nucleotides, P-Kmer - selected position-specific *k*-mers, comprising features from all considered values of k, P-All – composite positional features comprising selected features for P2, P3, P4, P5, and P6.



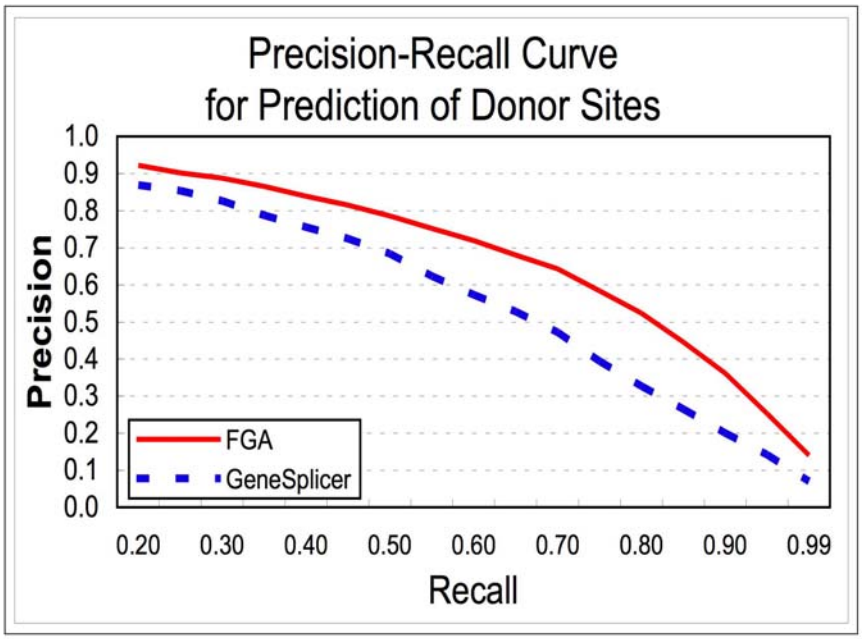
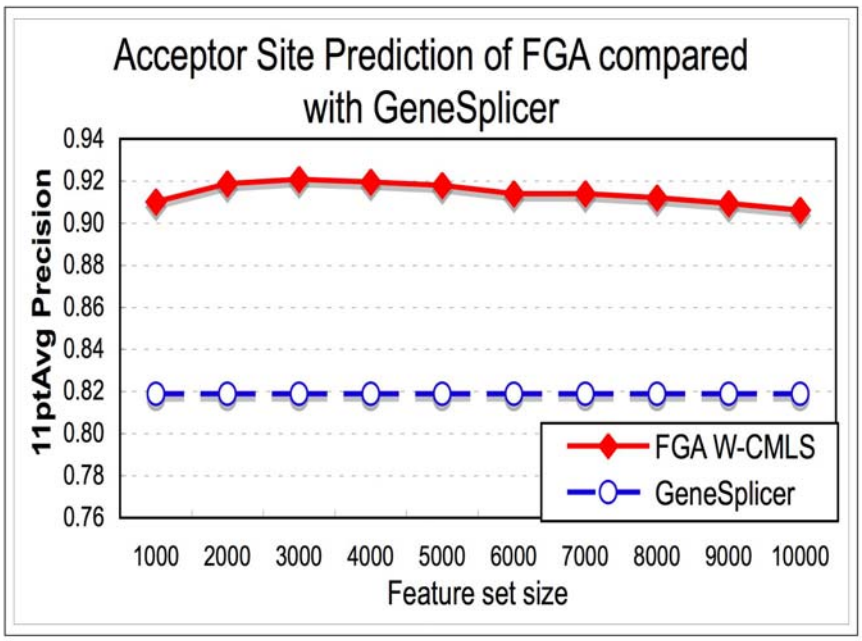


Figure 3.7. Precision results for each recall value for FGA with the complete set of features compared to GeneSplicer for acceptor and donor data.



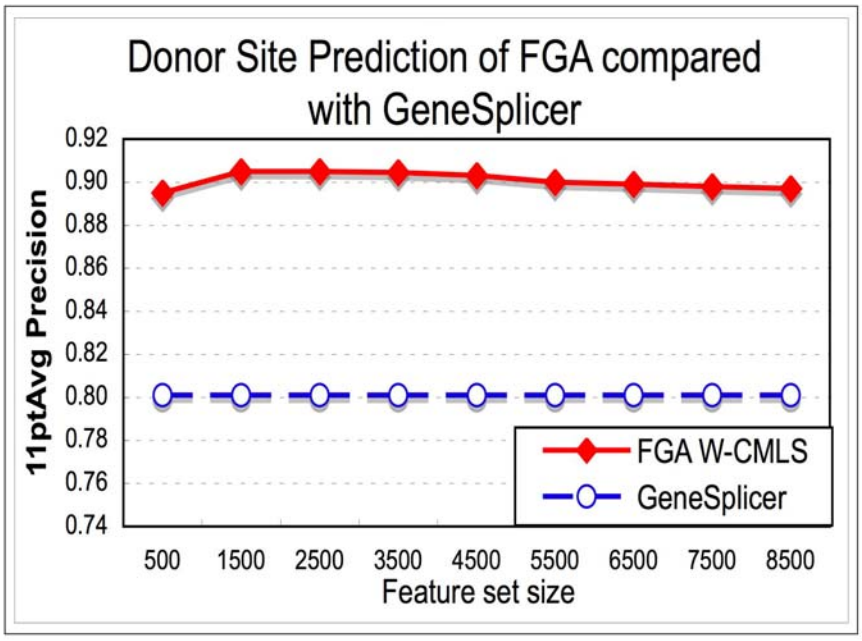
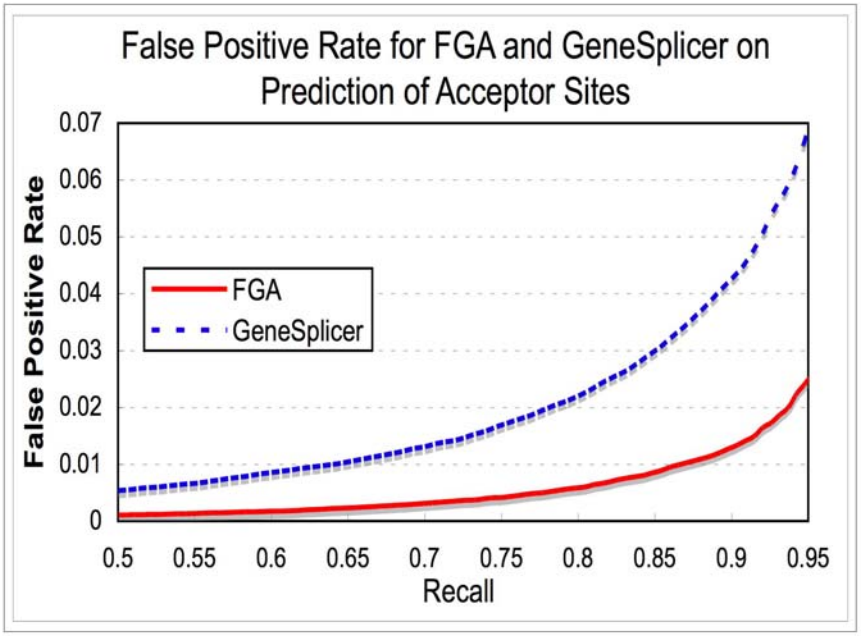


Figure 3.8. 11ptAvg precision results for FGA compared to GeneSplicer for acceptor and donor data.

We start with the complete set of features and recursively train the algorithm eliminating 1000 features at a time.



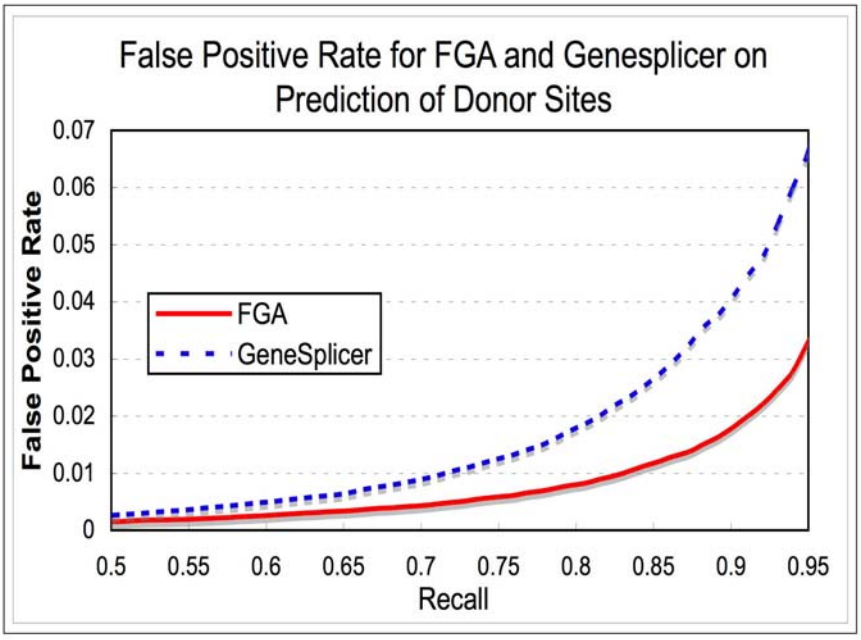
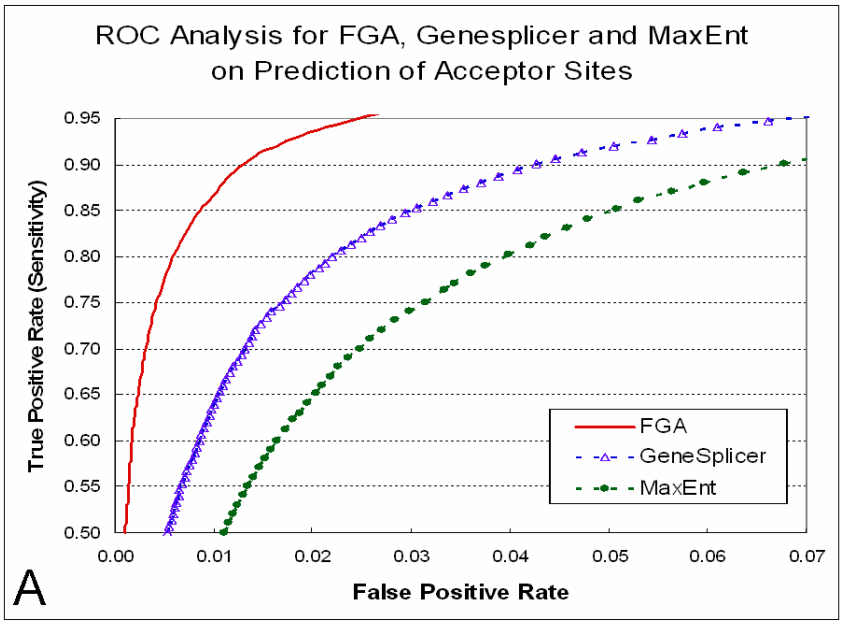


Figure 3.9. The false positive rate results for FGA with the final feature set compared to GeneSplicer, varying the recall threshold, for acceptor and donor data.



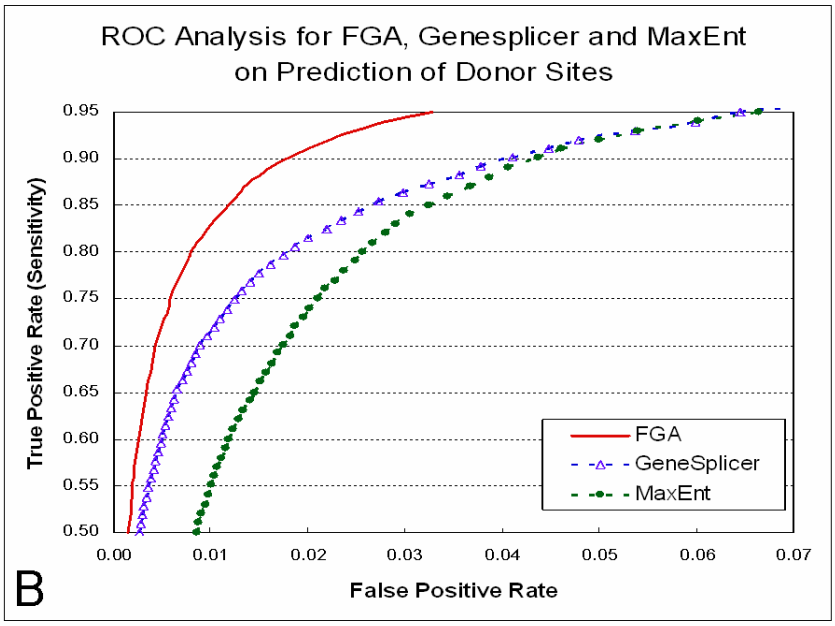
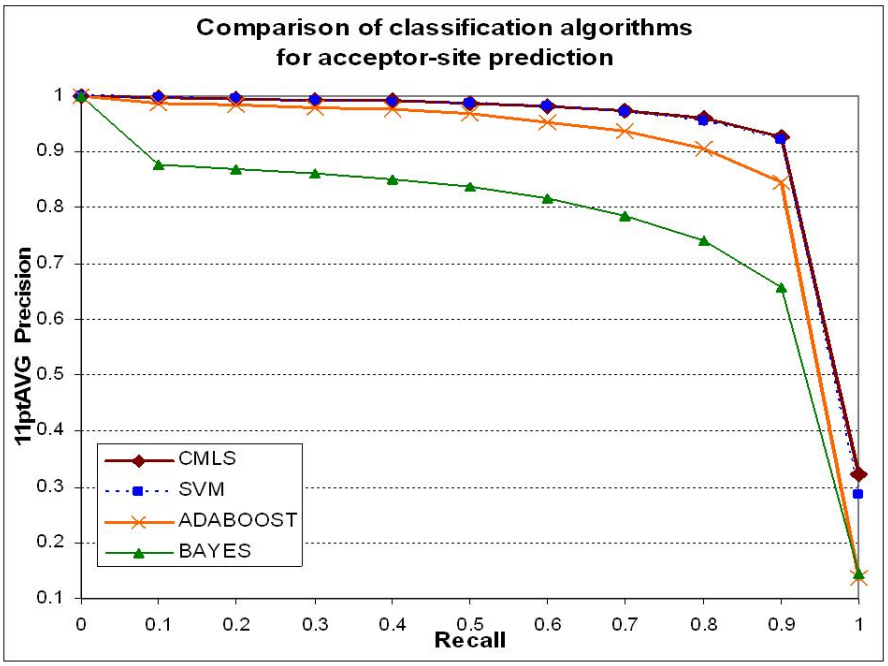


Figure 3.10. Receiver Operating Curve analysis for FGA, GeneSplicer, and MaxEnt for acceptor and donor splice-site prediction.

The true positive rate (TP/(TP+FN)) is plotted versus the false positive rate (FP/(FP+TN)). We show the sensitivity values ranging from 50% to 95%. When the score threshold for each method is adjusted, such that 5% of the true sites are missed (sensitivity is 95%), MaxEnt has recalled 10.48 % of the false sites; GeneSplicer, 5.80%; and FGA, only 2.49% for acceptor splice-site prediction, and MaxEnt has recalled

6.61 % of the false sites; GeneSplicer, 6.40%; and FGA, only 3.30% for donor splice-site prediction.These results are computed on the Human dataset of GeneSplicer, which contains 1115 pre-mRNA sequences.



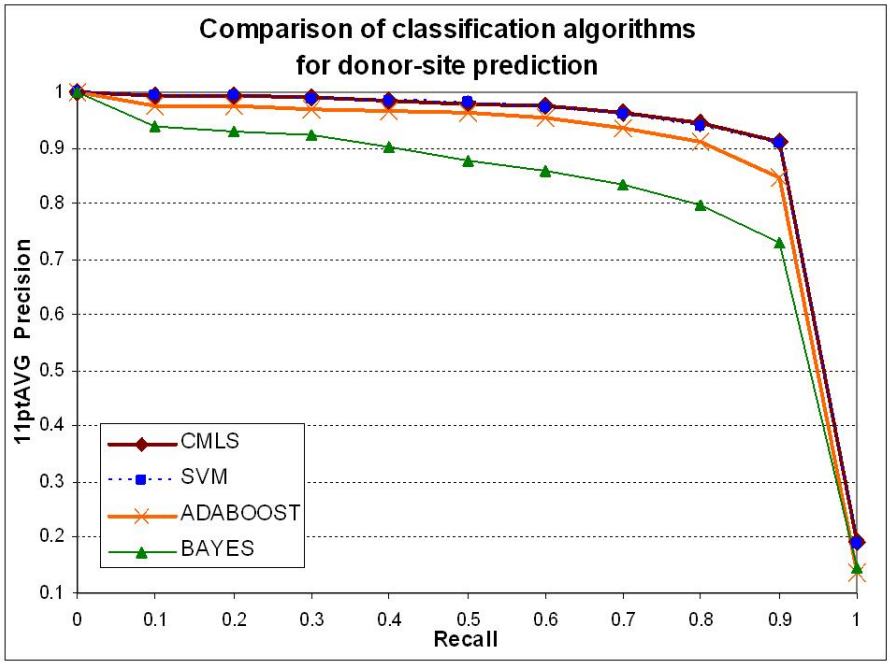


Figure 3.11. Precision-recall curve analysis for FGA, for acceptor and donor splice-site prediction comparing CMLS, SVM, AdaBoost and Naïve Bayes classifiers.

In all cases we performed a three-fold cross-validation, and here we present the average of the three folds. AdaBoost is run with decision trees as the weak classifier and the trees are grown until level 3. the least squares classifier, CMLS and SVM exhibit almost identical performances, with the exception that SVM took much longer to train. We decided to go with CMLS because of its speed and good performance.

Chapter 4: SplicePort — An interactive splice-site analysis tool

In Chapter 3, we described FGA, the feature-generation algorithm for sequence classification and the resulting splice-site prediction model that uses FGA-generated features. The FGA prediction model is capable of achieving high classification accuracy on human splice sites. The accurate selection of splice sites, as discussed in Chapter 2, requires both relatively well-characterized signals at the splice sites and auxiliary signals in the neighborhood sequence region. These signals are still not completely understood, and an easy-to-use method that would help biologists discover and interpret them is highly desirable.

In this chapter, we discuss our feature-space exploration to find biologically meaningful signals. In order to find relevant signals, we built SplicePort, a web server with rich functionality that is capable of predicting splice sites for user-input sequences, and browsing the whole collection of features generated by the FGA algorithm [150]. SplicePort may be important to a biologist for searching for interesting signals or validating previously observed signals that may be represented in the FGA-generated features. We discuss SplicePort in detail and we present examples detailing its rich functionality.

4.1 Discovering relevant splice-site signals

Accurate splice-site prediction is a critical component of eukaryotic gene prediction. Whole genome analysis of a single organism or comparison of genomes depends on accurate gene annotation. However, annotation is still limited by our ability

to properly identify splice sites [22]. As described in Chapter 3, FGA-identified features, in combination with a large-margin classification algorithm, produce accurate splice-site prediction on human pre-mRNA sequence data. These features capture important properties that distinguish actual splice sites from other similar DNA sequences, and they may help researchers in the further understanding of the splicing problem.

We have built SplicePort, a web-based interactive tool, which allows the user to explore the FGA features and allows the user to make splice-site predictions for submitted sequences based on these features. Other Internet resources that offer online splice-site prediction are: GeneSplicer [45], NetGene [25,7], MaxEntScan [59] and SplicePredictor [6]. For each input mRNA sequence, these web services provide the user with a list of predicted splice-site locations. However, a researcher may also be interested in identifying the signals used by the computational method to predict the splice site.

Any element in the DNA sequence of a gene that helps to specify the accurate splicing of the pre-mRNA sequence is a *splicing signal*. Branch sites, pyrimidine tracts, exon splicing enhancers, and silencers are all examples of known functional signals in the neighborhood of splice sites in eukaryotic genomes (see [35] for review). SplicePort, in addition to splice-site prediction, allows the user to explore all the FGA-generated features. None of the other online splice-site prediction systems provides this capability. We hope this will provide a useful resource for the identification of signals involved in specific splicing events, and possibly for the discovery of previously unappreciated splicing motifs.

4.2 SplicePort

The SplicePort web server is located at www.cs.umd.edu/projects/SplicePort or www.spliceport.org. From the SplicePort initial page, the user has two options: splice-site prediction and motif exploration. The splice-site predictor receives the user's input sequence and reports the whole set of predicted splice sites that confirm the constituent model (AG-dinucleotide consensus for acceptor and GT-dinucleotide consensus for donor). The motif explorer can be used to investigate acceptor and donor model feature sets identified in the input sequence or the sets of features FGA has discovered in the training sequences. The latter allows the user to browse the entire collection of positional features identifiable during the training phase. This motif exploration is novel and useful. While we illustrate its use on the FGA selected features, we believe this interface is general and can be used to explore other feature types [18,9,56] and features selected by other learning algorithms [21,65]. In Figure 4.1, we summarize the functionality of SplicePort, and we describe its components in greater detail in the following sections.

4.3 The FGA splice-site prediction model

We applied FGA to the task of splice-site prediction for the human genome (formally, the classification of AG dinucleotides into acceptors and non-acceptors and the classification of GT dinucleotides into donors and non-donors), as described in Chapter 3. FGA achieves very high accuracy compared to other splice-site prediction programs. For example, compared to GeneSplicer, FGA was able to achieve improvements of 43.0% and 50.7% in the reduction of the false positive rate, at the 95% sensitivity level, for acceptor splice sites and donor splice sites, respectively (see figures in Chapter 3).

As described in Chapter 3, for the human RefSeq training sequences, the FGA algorithm selected 3000 features for acceptor splice-site prediction and 1600 features for donor splice-site prediction. The acceptor site model contains 1362 compositional features and 1638 positional features, while the donor site model contains 764 compositional features and 836 positional features. We call these sets of features the acceptor-model feature set and the donor-model feature set.

The model feature sets then are used as input for the CMLS learning algorithm. For the splice-site prediction problem, two separate CMLS classifiers are required, one for acceptor, and one for donor sites. After the training phase of these classifiers, each feature f_i in the model feature sets is assigned a weight w_i. These weights define the decision boundary of the linear classifier that optimizes the performance. We also use these weights to derive feature ranking, as discussed in Chapter 3.

When the classification model is given a new input sequence (the sequence is in the format [80 nucleotides +AG/GT +80 nucleotides]), initially it checks whether it is a candidate acceptor (AG) or a candidate donor (GT) splice-site sequence. Then, the classifier checks the sequence if it contains any of the features previously identified by the FGA algorithm in the corresponding model feature set. The classifier produces a final score for the input sequence, adding the weights of each present feature. This score, assigned by SplicePort and displayed in the output, is best understood in terms of the splice-site classification problem itself.

In Figure 4.2, we use the B2hum dataset supplied by the GeneSplicer team to show the sensitivity and specificity differences for different FGA score thresholds. We

also provide a quantitative comparison between the two algorithms. Figure 4.2A depicts acceptor splice sites and Figure 4.2B depicts donor splice sites.

4.4 Splice-site prediction in SplicePort

Using the SplicePort splice-site predictor is straightforward. The user inputs a sequence in FASTA format. A sequence in the FASTA format is characterized by a header line starting with character '>', and containing a short description of the sequence. SplicePort also accepts sequences in a flat format. The sequence can be cut and pasted directly into the window, or uploaded as a separate file. The server is case insensitive and accepts either DNA (T) or RNA (U) sequences as input. The length of the submitted sequence determines the time required for prediction (approximately 1 second per kb of submitted sequence). Once the command to predict splice sites for the given input sequence is given, the system will use the FGA acceptor and donor model feature sets to score the given sequence. Each result is tested against the default score threshold (zero) and if it exceeds the threshold is displayed on the screen. Once the whole sequence has been processed, the user is able to download the complete set of results for personal records or parse the results using SplicePort as discussed below.

SplicePort splice-site predictor uses a splice-site neighborhood of 80 nucleotides upstream and 80 nucleotides downstream for a constituent splice-site. After the user submits the input sequence file, the results of splice-site prediction are displayed in a tabular format. Figure 4.3 shows a sample output. The table header includes: the input sequence description, which is the header line of the FASTA sequence; the sequence length; and, the sensitivity value and false positive rate for the current score threshold value for both acceptor and donor sites, which by default is set to zero.

For each prediction, the following information is listed: donor/acceptor splice site, the location in the sequence, a short subsequence centered at that location, and the FGA score. Predicted donor sites, occurrences of the dinucleotide “GT” in the input sequence, are listed in blue and predicted acceptor sites, occurrences of the dinucleotide “AG” in the input sequence, are listed in green. The location coordinate is measured from the first nucleotide in the input sequence. The short subsequence is 12-nucleotides long and is centered at “AG,” for predicted acceptor sites, and “GT,” for predicted donor sites. A threshold of zero, to our experience, is usually a good indicator that the predicted location has a high probability of being a true splice-site, however, the user can change the score threshold to increase or decrease the number of displayed predictions from the input sequence. Each new score threshold changes the sensitivity value and the false positive rate. The sensitivity value by default is 88.5% for donor sites and 88.8% for acceptor sites (correspond to score = 0). After each change, the new sensitivity and false positive rate values are calculated and displayed to the user, as shown in Figure 4.3(B). The user can choose to list all splice-site locations on the screen, or prefer to explore only donor or only acceptor, at a time, or switch between different views. In Figure 4.3(A), the user is listing all the sites, restricting the view only through score threshold. In Figure 4.3(B), the user has selected to explore only donor predictions and has changed the score threshold to 0.75. Please note that the sensitivity value and false positive rate have also changed accordingly. Finally, the user can select one of the predictions to investigate the identified signals, as described in the following section.

4.5 Browsing features on which a selected prediction is based

The characteristic that distinguishes SplicePort from the other online splice-site prediction tools is the capability to explore the features identified in any prediction on the original input sequence. SplicePort allows the user to explore potential splicing signals in the vicinity (160 nucleotides) of any particular splice site (AG or GT), by examining the features that contribute to the score assigned to that potential site. The signals of the acceptor model feature set or the donor model feature set can be listed, browsed, and visualized by selecting the Browse Features option.

Features are grouped into compositional features and positional features. Usually, a subset of acceptor or donor model feature set is present in any predicted splice site. Compositional features comprise general, upstream and downstream *k*-mers. They can all be listed, clustered and sorted by their weight. Positional features comprise position-specific nucleotides, position specific *k*-mers and conjunctive n-positional features in the 160-nucleotide neighborhood. There are a variety of browsing possibilities for this set of features. The user specifies an interval within the 160-nucleotide window by giving the starting and the ending points. All the positional features associated with positions within this interval are listed. They are shown relative to the splice-site location, providing the user with a visual representation of the position of the feature, and are ordered by the absolute value of their individual weights. The user may further group these features, draw histogram and WebLogo [15] frequency plots, search by motif, and set the weight threshold.

As an example (see Figure 4.5), we used SplicePort to examine exon 7 of the homologous SMN1 and SMN2 genes, a well-studied case [9] where a single nucleotide

difference at position 6 of the exon (C→T) accounts for reduced inclusion of this exon in SMN2 (see [9] for review). The SMN gene is linked with a common human genetic disorder called the Spinal Muscular Atrophy (SMA). SMA is a motor neuron disease. The motor neurons affect the muscles that are used for activities such as crawling, walking, head and neck control, and swallowing. Researchers have identified the SMN1 gene (survival motor neuron 1) as the primary manufacturer of the SMN protein. It is the absence/defect of this SMN1 gene that causes SMA. An individual with SMA has a missing or mutated gene that does not produce as much protein, or the right kind of protein. Since SMN protein is critical to the survival and health of motor neurons, without this protein nerve cells may atrophy, shrink and eventually die, resulting in muscle weakness.

SplicePort scores the SMN1 exon 7 acceptor and donor 1.78 and 0.02, respectively and the single nucleotide change in SMN2 reduces these numbers to 1.61 and -0.18. This difference is very subtle for the acceptor site but the change in the donor site score is enough to increase the false positive rate from 1.34% to 2.08%. This means that the single nucleotide mutation causes this donor site to be harder to recognize, which may be the reason that this exon is sometimes skipped. SplicePort feature browser shows that the difference in donor scores is primarily due to the negatively scoring upstream feature TAG (-0.18).

The seventh exon of the SMN gene is 54 nucleotides long. The single nucleotide mutation occurs six positions downstream the acceptor splice site and 48 positions upstream the donor splice site. Most of the splice-site predictors would give exactly the same splice-site score for both cases (i.e. MaxEntScan). GeneSplicer and NetGene would

pick up the difference because they look at a wider splice-site neighborhood, similar to SplicePort, however, they would not be able to point to the features that cause the difference in scores.

4.6 Motif exploration tool

Users can explore general features discovered by FGA for human RefSeq sequences, using the motif exploration tool. The sequence-specific feature browser shows only those features used to score the submitted sequence (from the acceptor or donor-model features sets). In contrast, motif exploration tool presents a much richer set of features. In order to facilitate motif discovery, rather than focusing on the simple compositional features, here we have made available a variety of positional features as selected through several iterations of FGA. These features are much richer than the features of existing splice-site tools. Each composite positional feature set we considered is the conjunction of a *k*-mer and a number of arbitrary position-specific nucleotides. We denote a specific set using the notation Kmer+X; for example, 4mer+2 is the set of 4mers together with two position-specific nucleotides.

Figure 4.6 illustrates a portion of SplicePort motif explorer. Acceptor and donor-site features are grouped in two conceptually similar interfaces. The figure on the top shows how the user selects a feature set and specifies an interval to browse the features. The figure on the bottom shows the results. In this example, the user is exploring the features generated for acceptor splice-site prediction.

The features are shown with respect to the splice-site location, and they are ordered according to the absolute value of their weight. The weight of a feature is learned by the CMLS classification algorithm during training. These weights can be used to order

and group the features. A positively weighted feature is a feature mostly found in splice-site sequences, and a negatively weighted feature is a feature more commonly found in non-splice-site sequences. Figure 4.7 shows the results of WebLogo and Histogram functions. The user can view a depiction of the positively and negatively weighted features in the specified interval by generating a WebLogo frequency plot. The histogram allows the user to visualize the role of each nucleotide for each position in the specified interval. We represent this with four different bars, one for each nucleotide, for each position. The height of each bar is the accumulated weight for that position-specific nucleotide and is calculated using the weights of all the features that have that nucleotide at that position.

Because the features generated with the FGA algorithm are position-specific features, we may find the same pattern of nucleotides repeated in a given interval. Interval Features refer to a set of features that share the same pattern of nucleotides but differ in starting positions. The user can list all the interval features for a specified interval and feature set. SplicePort displays the number of individual features as well as their average weight. To obtain the list of all individual features shown relative to a splice site in their respective locations, the user can use the Search by Motif option. This option also facilitates the search for known motifs or partial motifs. The user enters a short sequence and receives a list of all features in the specified interval that contain that sequence.

In addition, for each feature set and specified interval we perform a clustering procedure based on edit distance. We identify similar features, and the tool groups them together generating WebLogo frequency plots to represent them. The user can browse

these identified clusters and their individual elements by selecting Identified Motifs. This option may help users identify known functional motifs and may guide them in the search for new ones.

An illustrative example inspired by the case of SMN1 and SMN2 is a comparison of TAG and CAG among 5mer features located in the -60 to -30 interval relative to donor sites. Features containing TAG are all negative, with multiple examples of TTTAG. Conversely, CAG shows primarily positive features. This example is shown in Figure 4.8. Additional examples of using the SplicePort motif exploration tool are described in the Chapter 5, where we describe finding biologically relevant motifs in FGA features.

4.7 Summary

The SplicePort server is a versatile tool with two main functions. First, the user can perform accurate splice-site prediction on a sequence, which they input to the tool. Splice-site prediction has the added flexibility of exploring all the putative splice-site locations, their score, corresponding sensitivity, and false positive rate values. Second, the user can explore the motifs for the requested location in the input sequence and browse the complete collection of identified motifs for both acceptor and donor splice sites. This tool can both help a user decide whether there is a splice site in the given sequence and also allow the user to identify elements of functional motifs. An additional benefit of a computational exploration approach, such as SplicePort, is that it can be readily implemented in other genomes.

In summary, SplicePort allows the user to gain useful insight in gene splicing signals. This data analysis tool provides the community of researchers investigating pre-mRNA splicing with a powerful and flexible resource for the identification of functional

elements. Motif exploration enables researchers to rapidly explore the space of computationally identified signals and effectively pose hypotheses for experimental test and validation.

4.8 Figures of Chapter 4

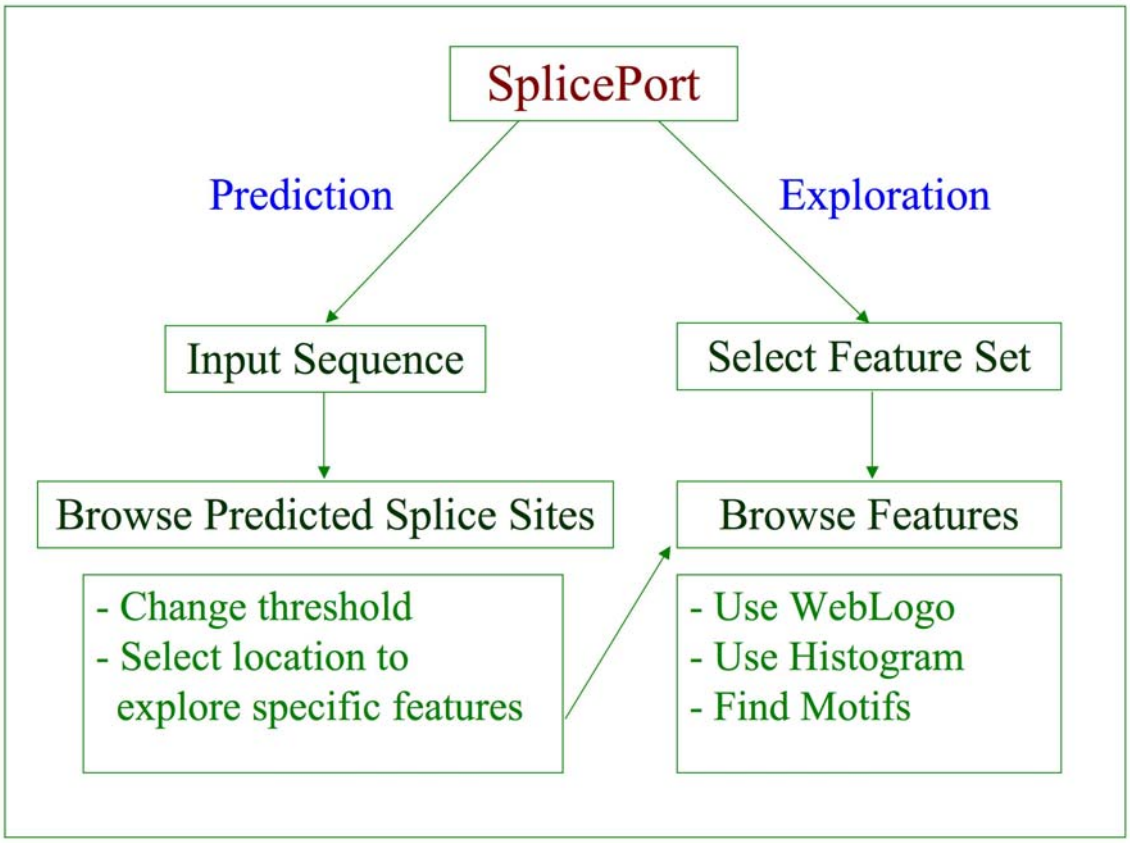


Figure 4.1. Organization of the SplicePort interactive interface.

On the starting page, a user chooses between splice-site prediction and motif exploration. After potential splice sites are predicted and scored, the features on which the predictions are based can then be explored.

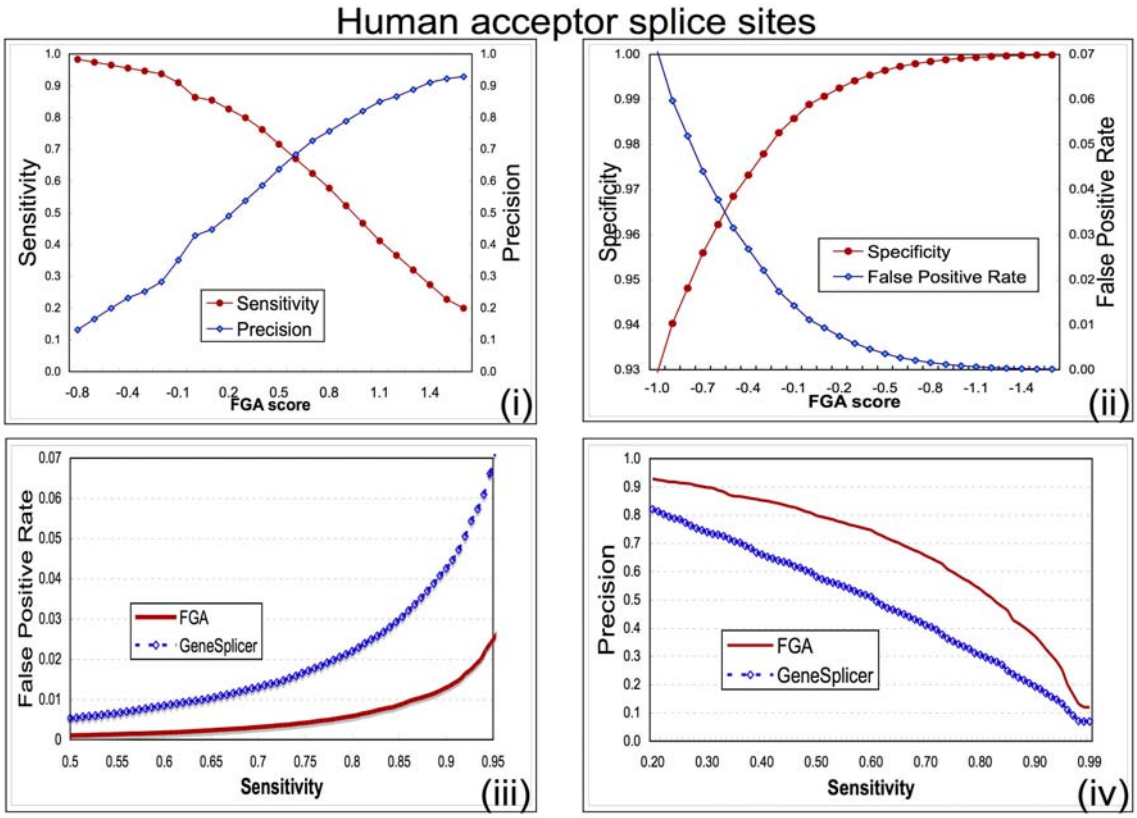


Figure 4.2A. Splice-site predictor for human acceptor sites.

Part (i) depicts the sensitivity, TP/(TP+FN), and Positive Predictive Value, TP/(TP+FP), also known as precision, vs. FGA score for the human acceptor splice sites. Part (ii) depicts specificity, TN/(TN+FP), and False Positive Rate, FP/(TN+FP), vs. FGA score. Figures (iii) and (iv) show the False Positive Rate vs. Sensitivity and Precision, vs. Sensitivity. In Figures (iii) and (iv) FGA results are compared with those of GeneSplicer. These results show that FGA produces fewer false positives and higher precision for every sensitivity threshold. These differences are highly statistically significant.

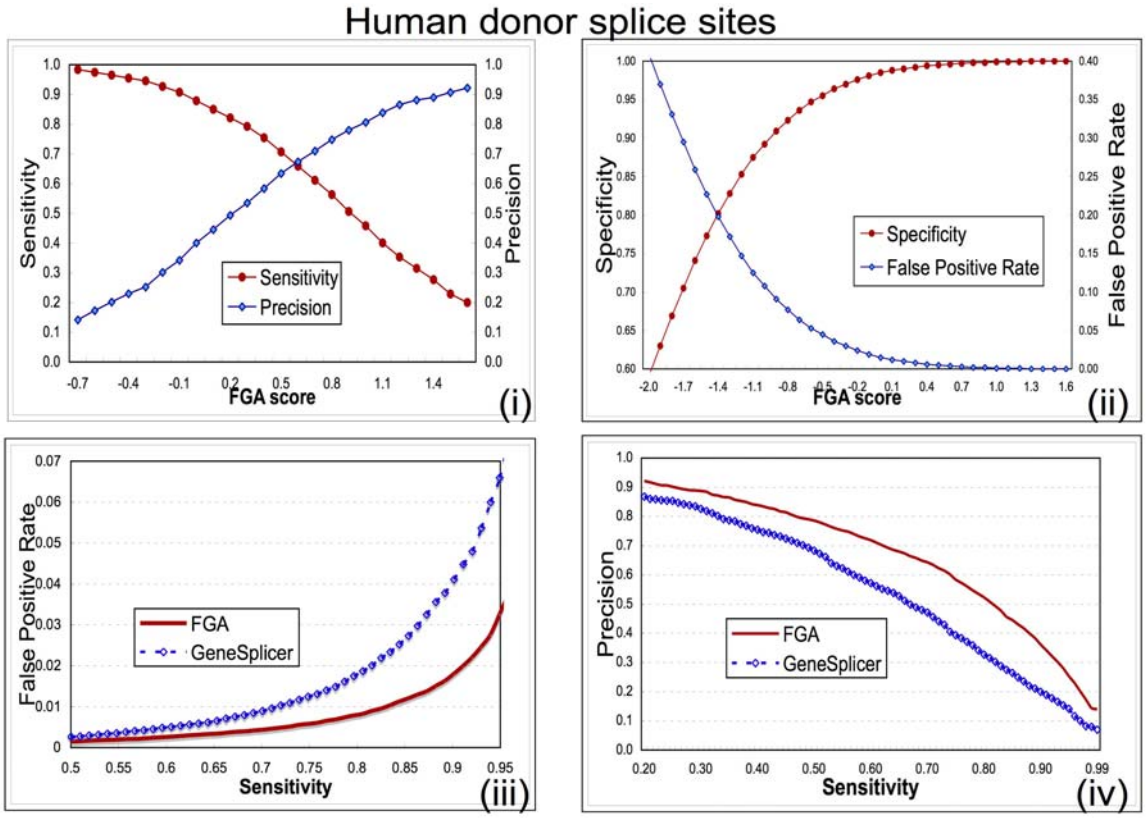


Figure 4.2B. Splice-site predictor for human donor sites.

Part (i) depicts the sensitivity, TP/(TP+FN), and Positive Predictive Value, TP/(TP+FP), also known as precision, vs. FGA score for the human donor splice sites. Part (ii) depicts specificity, TN/(TN+FP), and False Positive Rate, FP/(TN+FP), vs. FGA score. Figures (iii) and (iv) show the False Positive Rate vs. Sensitivity and Precision, vs. Sensitivity. In Figures (iii) and (iv) FGA results are compared with those of GeneSplicer. These results show that FGA produces fewer false positives and higher precision for every sensitivity threshold. These differences are highly statistically significant.

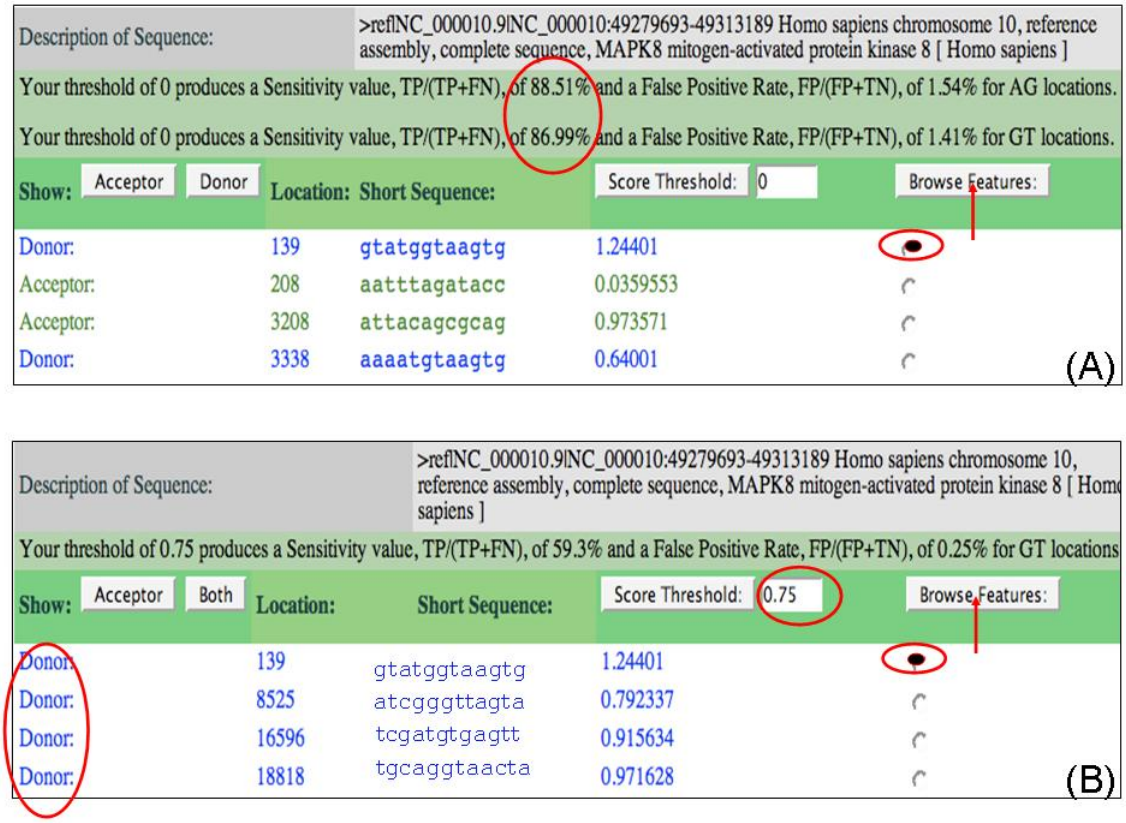


Figure 4.3. Splice-site predictor in SplicePort.

Part (a) depicts a typical output example of the predicted splice sites. We have circled the displayed sensitivity value. From this screen, the user can select a predicted site, we have selected the donor site at location 139 for illustration, and click on Browse Features, which we show with the arrow, to explore the present features.

Part (b) depicts the situation when the user prefers to explore acceptor or donor splice-site locations separately. The user can browse the features that are present in the checked sequence by clicking on Browse Features, which we show with the arrow. The user can change the score threshold, which we have circled on this screen, and list all the sites that score higher than the threshold. The sensitivity and false positive rate values are shown below the FASTA sequence description line.

SMN1 : -------GGTTTCAGACAAAATCAAAAAGAAGGAAGGTGCTCACATTCCTT----------

Description of Sequence:		>reflNC_000005.8lNC_000005:70282524-70284577lSMN1exon7with1kbflanks		
Your threshold of -0.5 produces a Sensitivity value, TP/(TP+FN), of 96.65% and a False Positive Rate, FP/(FP+TN), of 4.34% for AG locations.				
Your threshold of -0.5 produces a Sensitivity value, TP/(TP+FN), of 95.54% and a False Positive Rate, FP/(FP+TN), of 3.85% for GT locations.				
Show: Acceptor Donor	**Location:**	**Short Sequence:**	Score Threshold: -0.5	Browse Features:
Acceptor:	1000	cttacagggttt	1.78077	
Donor:	1054	aaggagtaagtc	0.0200066	
Acceptor:	1498	tttgcaggaaat	0.665236	
Donor:	1640	aatgggtaactc	0.243022	

(A)

SMN2 : -------GGTTTTAGACAAAATCAAAAAGAAGGAAGGTGCTCACATTCCTT----------

Description of Sequence:		>reflNC_000005.8lNC_000005:70282524-70284577lSMN1exon7with1kbflanks		
Your threshold of -0.5 produces a Sensitivity value, TP/(TP+FN), of 96.65% and a False Positive Rate, FP/(FP+TN), of 4.34% for AG locations.				
Your threshold of -0.5 produces a Sensitivity value, TP/(TP+FN), of 95.54% and a False Positive Rate, FP/(FP+TN), of 3.85% for GT locations.				
Show: Acceptor Donor	**Location:**	**Short Sequence:**	Score Threshold: -0.5	Browse Features:
Acceptor:	1000	cttacagggttt	1.6057	
Donor:	1054	aaggagtaagtc	-0.177324	
Acceptor:	1498	tttgcaggaaat	0.665236	
Donor:	1640	aatgggtaactc	0.243022	

(B)

Figure 4.5. Splice-site prediction output of SplicePort for SMN gene.

SMN1 exon 7 gene sequence is shown in part (A), and SMN2 in part (B), with 1kb nucleotides flanking region in both cases. The acceptor site of exon 7 is at position 1000 and the donor site is at position 1054. We see that the single nucleotide difference at position 6 of the exon reduces the acceptor score from 1.78 to 1.61 and the donor score from 0.02 to -0.18.

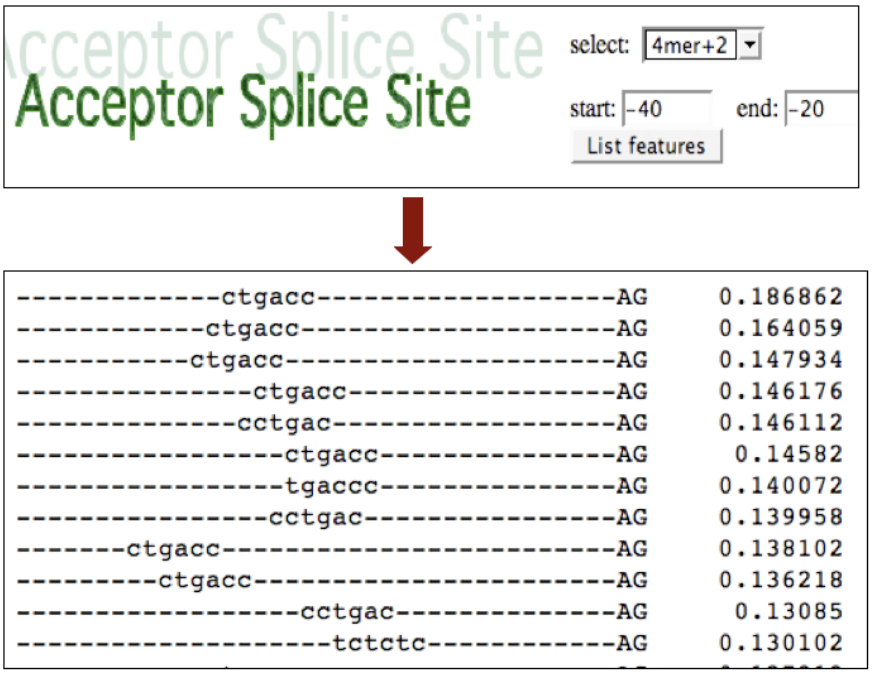


Figure 4.6. Motif Exploration Tool in SplicePort.

This figure shows initially the selection of the feature set 4mer+2 in the branch site interval. SplicePort outputs the list of features in the specified interval. Each feature is aligned to the splice site position and has a weight assigned to it by the FGA algorithm. The acceptor splice site is depicted in the output with the capitalized di-nucleotide AG.

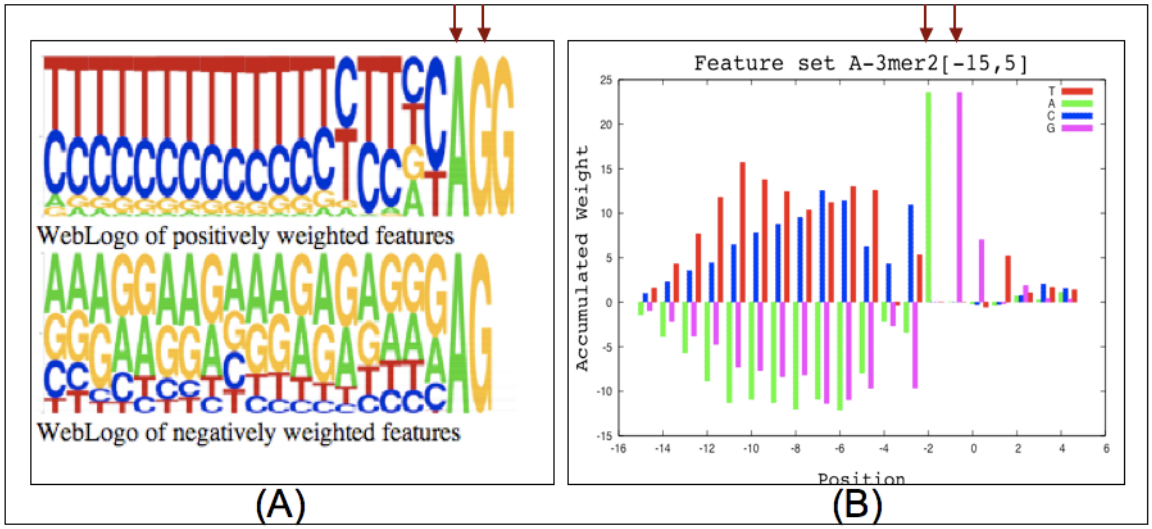


Figure 4.7. Typical outputs of motif exploration in SplicePort.

The are outputs of FGA features for acceptor splice-site prediction: part (A) shows WebLogo frequency plots of features when we select the interval [-20,1], and part (B) shows the histogram generated from accumulated weights of features when we select the interval [-15, 6]. The small arrows denote the location of acceptor splice-site consensus dinucleotide AG.

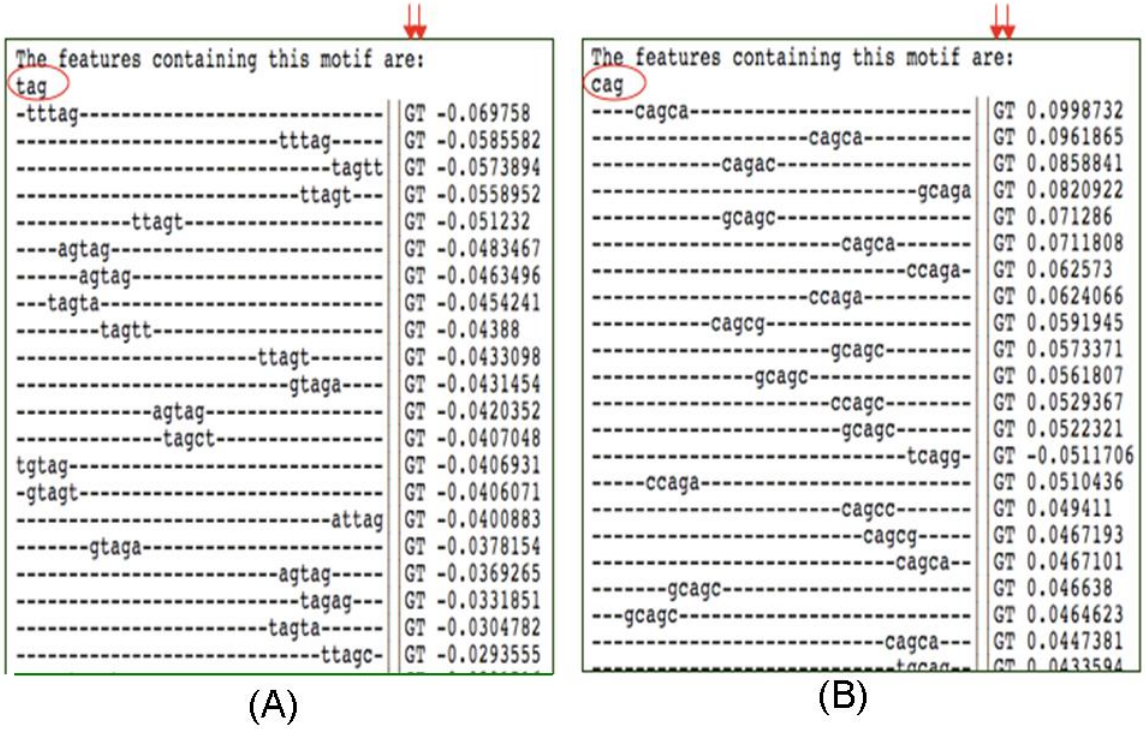


Figure 4.8.Outputs of SplicePort motif exploration for SMN gene related features.

Outputs for 5mer feature set of donor splice-site prediction in the selected interval [-60,-30], related with the SMN1 exon 7 example. On the left, we list features that contain the motif “tag.” Note that all these features have a negative weight. On the right, we list features that contain the motif “cag.” Note that these features are mostly positive. The small arrows denote the location of donor splice-site consensus dinucleotide GT.

Chapter 5: Features generated for splice-site prediction correspond to functional elements

In general, *knowledge discovery* is the analysis of the data to find patterns and models that help summarize the data in novel ways that are both understandable and useful to the data analyst. A supervised machine-learning algorithm uses a set of known examples (the training set) and a set of characteristics or *features* describing the training set to construct a model of the data. The learned model is evaluated by testing its accuracy on a held-out test set. An important input to any machine-learning algorithm is the choice of features describing the dataset. A challenge, which we have addressed with the feature-generation algorithm, is how to determine the best set of features for a given prediction task. Another challenge, which we have addressed in Chapter 4 with SplicePort, is how to discover, interpret, and assess the identified features.

In this chapter, we explore the knowledge-discovery power of the FGA algorithm by taking a closer look at the generated features, using the motif exploration tool of SplicePort. We present examples of the observed feature groups and describe our efforts to detect biological signals that may be important for the splicing process. We find that the features generated for computational splice-site prediction include known functional elements, and we present evidence that these features provide previously unknown information about some aspects of these splicing signals.

5.1 Description of FGA feature sets in SplicePort

Here, we summarize the specific steps used to generate the composite feature sets used in the motif exploration tool in SplicePort. As we already discussed in Chapter 3, a composite positional feature set is the conjunction of a *k*-mer and a number of arbitrary position-specific nucleotides. To generate a composite positional feature set, we need to specify an initial set of features, an appropriate construction method, and a fast feature-selection method. To prepare the initial sets of features, for both donor and acceptor splice-site prediction, we started with the position-specific *k*-mer sets for *k* from 3 to 6. The numbers of potential features for these feature sets are, respectively, *10,240, 40,960, 163,840,* and *655,360*. For each of these sets the Information Gain feature-selection method was used to select the top scoring 5000 features. These sets constituted our initial feature sets for the construction algorithm. As described in Chapter 3, the feature-construction method expanded each of these sets by adding one position-specific nucleotide in an unconstrained position. After the construction step, we again used information gain to evaluate each of the features in the constructed set. Then we evaluated each feature according to a logistic scheme, taking into account the distance between the newly added nucleotide and the original feature, preferring features for which the distance was smaller.

After the feature selection step, the top scoring 5000 features were selected. These sets constituted the input sets for the next iteration. We ran the algorithm and generated features up to, at most, 10 conjunct nucleotides in different positions in the composite feature sets. For each set of features we built a separate splice-site prediction model using the CMLS classification algorithm. Table 5.1 summarizes the

splice-site prediction performance for each of these feature sets. Some of these sets performed better than others, but in our analysis we explored all the sets for the purpose of knowledge discovery.

In what follows, we use the shorthand notation $S-kMERn[p_1,p_2]$ to describe the composite feature subsets that we studied. In this notation, $S \in \{A,D\}$ stands for acceptor (A) or donor (D) splice sites, kMER stands for the number of consecutive position-specific nucleotide features in the initial set, *n* is the number of additional conjuncts and [*p1*,*p2*] denotes the interval from position *p1* to position *p2* in the sequence. For example, $A-3mer3[20,40]$ is a subset of acceptor splice-site features. These features were generated from the initial set of position-specific 3-mer features and were obtained after three FGA iterations, adding each time a new nucleotide in an unconstrained position within the specified interval. The sequence positions associated with each of the features in this subset were from the coding region 20 to 40 nucleotides downstream the acceptor splice site.

Following with our definitions, we say that two composite features match if they share the same nucleotide pattern, starting at different positions. For example, let $4mer[1,10]=\{a_1g_2c_3t_4, a_6g_7c_8t_9\}$ be the subset of composite 4-mer features from the interval [1,10], where a_1 denotes nucleotide a at the first sequence position. In this case, the features $a_1g_2c_3t_4$ and $a_6g_7c_8t_9$, are two matching composite features. A composite feature subset may contain several matching features that differ only in the starting position within the specified interval. We represent a set of such occurrences with an interval-feature pattern, e.g. $a_ig_{i+1}c_{i+2}t_{i+3}$. An interval-feature pattern is the nucleotide pattern shared among the matching composite features and the number of

interval occurrences of a feature pattern is the number of matching composite features it represents. We use the notation $S-kMERn[p_1,p_2]^*$ to denote the set of all interval- feature patterns for the subset $S-kMERn[p_1,p_2]$. For the above example, given the set of features $4mer[1,10]=\{a_1g_2c_3t_4, a_6g_7c_8t_9\}$, the set of interval-feature patterns is $4mer[1,10]^*=\{a_ig_{i+1}c_{i+2}t_{i+3}\}$. The number of occurrences for the pattern $a_ig_{i+1}c_{i+2}t_{i+3}$ in the given feature set is two.

In our analysis, features were ranked according to the weight assigned to them by the classification algorithm. We used the WebLogo program to draw frequency plots. We plotted histograms and used basic *k*-means clustering algorithms and edit-distance measures to cluster the features into groups. Here we list some of our findings and illustrate them with our features.

5.2 Knowledge discovery: generated features capture biological signals

What kinds of biological signals do these generated features capture? Examples of positive signals that we might expect to find in a typical pre-mRNA include the branch site, the pyrimidine-rich region close to the acceptor splice site, splice-site consensus signals themselves, and exonic splicing enhancers. In addition, it is likely that sequence elements associated with the coding sequence are present among our features. One may ask whether the signals identified through the exploration of the features positioned on the exonic regions of the sequence are really splicing signals or, in fact, are signals that reflect the coding properties of exons. Admittedly, not all features can be splicing signals. However, at the core of these features lies the generation procedure described with the feature-generation algorithm. And FGA has identified these features when learning the difference

between annotated splice-sites and randomly picked AG/GT surrounding regions (pseudo-splice sites). We believe that these features do not emphasize the coding properties of exons and we are mentioning two arguments in support of our claim. First, recall the composition of our training sequences. The set of our negative training sequences consists of randomly picked AG/GT locations in the original pre-mRNA sequences and their surrounding regions. The only constraint we imposed on the AG/GT locations is that they should not be annotated splice-site locations. Therefore, many of our negative training sequences overlap with exons, and features capturing coding properties are expected to be present in our negative training sequences, although, admittedly, in a non-dominant level. The positive training sequences consist of annotated splice-site sequences from all the coding exons in the original pre-mRNA sequences. So, it would be interesting to know how would FGA splice-site predictor on identifying splice-sites that belong to non-coding exons. And this forms our second argument. Second, and most significantly, FGA performs well on introns flanked by non-coding exons. We compiled all the annotated splice-site sequences flanking non-coding exons in our original pre-mRNA sequences. There were 4961 acceptor splice-site sequences and 2148 donor splice-site sequences fulfilling these criteria. We used the same set of negative sequences and we tested the performance of FGA on the classification of splice –sites in this new set. Our results show that FGA shows an 11ptAVG of 83.33% for acceptor sites and 64.52% for donor sites. These are impressive results for recognition of non-coding exons flanking splice sites. So, our features are in fact predominantly capturing splicing signals.

5.2.1 The Branch-Site interval

The mammalian branch-site signal is difficult to describe because it is degenerate and shows very low levels of purifying selection [31]. The branch site plays a central role in the chemistry of splicing. In the first step in splicing the branch site reacts with the upstream exon. The consensus is TNCTRAC [42], although this sequence is based on very few biochemical determinations compared to the splice-site sequences, and is poorly conserved. The **A** is the base that attaches to the donor splice site, and it is usually located from 18 to 40 nucleotides upstream of the acceptor site, although it can be much more distant.

In order to investigate the branch-point signal, we examined composite features of 6 nucleotides that start in the interval from 40 to 20 nucleotides upstream from the acceptor splice site (and therefore extend from -40 to -15). Our current feature set for this purpose was $A-3mer3[-40,-20]$. The subset contained 346 selected features.

Table 5.2 shows the top-scoring 20 features in their exact position with respect to the annotated acceptor site, which is found 15 nucleotides downstream of the interval shown. Each feature is listed, ranked by the weight assigned by the CMLS classification algorithm. A large number of positional features in this feature set captured the branch-point signal. In fact, of the 30 features that had weights above 0.1 in this set, all but 5 contained either CTRA or at least five pyrimidines. In absolute numbers, 97 individual features of this set matched the branch-point consensus and 158 features were pyrimidine-rich. The rest of the features were assigned negative weights. The negatively weighted features comprised a G-rich

signal mostly. Of those, 44 features matched the pattern AGG and the others were A-rich.

Table 5.3 illustrates a subset of $A-3mer3[-40,-20]*$ interval-feature patterns. Each listed pattern represents at least five matching composite features, differing only in the starting position in this interval. The number of interval occurrences is also given and an average weight is computed for each interval-feature pattern from the individual CMLS weights assigned to the distinct matching composite features during training. We grouped these patterns into three categories: 1) nine interval-feature patterns matching the branch-site consensus, 2) two pyrimidine-rich interval-feature patterns, and 3) two negatively weighted purine-rich interval-feature patterns.

Table 5.4 lists all the position-specific occurrences of GCTGAC in the [-80, -1] interval. These features matched the branch-site consensus and they were assigned positive weights by the classification algorithm. The distribution of scores for this one hexamer suggests a preferred location for the branch site A at -30 to -20. Many independent observations with related features (e.g. CTAAC) indicated a similar region. For example, in Figure 5.1, we present a comparison of four tetramer features present in the set. It is apparent from the distribution of these features that positions -27 through -16 are preferred for the branch site A. This observation agrees well with experimental results [12].

5.2.2 The acceptor splice-site (pyrimidine-tract) interval

The protein factors that recognize splice sites need to bind to a variety of sequences. An example is the structure of U2AF65, or PUF60 which binds to the

polypyrimidine tract. In Figure 5.1 we also show the distribution of TTTT and CCTT, in this interval. Note that this distribution is broader than the distribution of branch-site tetramers. In addition, there is a region (-16 to -12) where the scores assigned to TTTT become negative and tetramers containing C have maximal scores. Similar peaks are observed for CTTT, TCTT, TTCT and TTTC, as shown in Table 5.5.

In order to further investigate the characteristics of the upstream region close to the acceptor splice site, we also examined the feature set $A-5mer[-20,-1]$. There were more than 2,000 selected features in this subset. We note that a large number of features were selected in this set, indicating stronger potential signals close to the splice site. Based on the weight assigned by the CMLS algorithm, we divided these features into two groups; positively weighted features and negatively weighted ones. In Figure 5.2, we used the WebLogo program to draw a frequency plot of the two groups of features. The annotated acceptor site is shown in the figure with the consensus dinucleotide AG.

One interpretation from these plots is that the generated features are capturing the pyrimidine tract, and that they are scanning along the sequence for the exact AG dinucleotide consensus where the true acceptor site is located. The difference between the two frequency plots for positively and negatively weighted features is striking. Figure 5.2a shows that the presence of the CT-rich feature is very important in this interval and Figure 5.2b shows that the presence of an AG-rich element is an indicator of a non-splice sequence. The frequency plot for the positively weighted features (Figure 5.2a) is very similar to the acceptor splice-site consensus itself. However, our features do not simply reflect the nucleotide frequencies seen at true

sites. Figure 5.2c and Figure 5.2d show the frequency distribution of the true acceptor sequences and non-acceptor sequences in the training dataset. The frequency distribution of the non-acceptor sequences in our dataset in the pyrimidine-tract interval (Figure 5.2d) is different from that of the negatively weighted features in the $A-5mer[-20,-1]$ feature set (Figure 5.2b).

In other words, our features were better than frequency data alone at discriminating true splice sites. To illustrate this difference, we used the frequency distribution matrices of these data to discriminate the true splice sites, achieving an 11ptAvg precision of 40.1%. On the other hand, when we trained a CMLS classifier on the FGA feature set, it achieved an 11ptAvg precision of 80.6% for the same task.

Exploring the pyrimidine-tract interval further, we selected another feature set, which was characterized by composite positional features containing 7 nucleotides in different positions, $A-6mer1[-20,-1]$. We made a list of the features, and we identified clusters of similar features, using the *k*-means clustering algorithm with the edit-distance similarity measure. Figure 5.3 shows some examples and samples of the features in each group.

5.2.3 GGG motifs near the 5' slice site

In order to investigate the characteristics of introns near the 5' splice site, we explored the intron downstream of the 5' splice site, using a number of parameters. In each case, GGG and GGGG motifs were common. For example, the $D-3mer3[6,64]$ set included 54 positively ranked occurrences of GGG and four negatively ranked occurrences. A plot of scores versus position for GGG and GGGG is provided in Figure 5.4A and Figure 5.4B, showing that this motif scores positively in the intron

downstream of 5' splice sites but negatively in the flanking exon. GGG likewise dominates $D-3mer3[-80,-40]$. A number of papers have reported a role for GGG and GGGG motifs in splicing [23,33,40]. Recognition of these motifs has been attributed to the U1 snRNP [39] and hnRNP H [23].

5.2.4. The donor splice-site interval

Next, we investigate the characteristics of the donor splice site. Sample clusters, similar to those created for the acceptor site, are shown in Figure 5.5. The first two sequence logos, Figure 5.5a and Figure 5.5b, show the frequency plot of the positively and negatively weighted groups of features for the set $D-6mer[-10,10]$. The donor splice-site consensus sequence is MAGGTRAGT (where M is A or C and R is A or G). The next two plots, Figure 5.5c and Figure 5.5d, show the frequency plot for the same interval based on the true donor and non-donor sequences in the training dataset. Once again, the sequence logo of the positively weighted features resembles the logo of the nucleotide frequency of the positive data, but important differences are apparent, especially at positions on the periphery of the region shown.

5.3 Exon Splicing Enhancers (ESEs) and Exon Splicing Suppressors (ESSs)

We also compared our generated features to published work on Exonic Splicing Enhancers (ESEs) and Exonic Splicing Silencers (ESSs). ESEs and ESSs are short oligonucleotide sequences located in the exonic region that affect splicing. The presence of ESE sequences in the exonic region results in the enhancement of the recognition of the nearby splice sites. The presence of the ESS sequences, on the other hand, suppresses nearby splicing events. These regulatory signals have been

studied experimentally (reviewed in [68]) and computational methods have been built to find them [9,18,56,21,46,52].

We considered the set of distinct hexamers in the flanking exon interval, for both acceptor and donor by computing interval features of the region of the sequence downstream from the annotated splice site for acceptor and upstream for donor. We divided this set of interval features into positively and negatively weighted sets. We compared these sets of hexamers with a list of experimentally identified ESE's and ESS's of mammalian and viral RNA [68]. There are 61 experimentally determined ESE sequences listed by Zheng [68], including some that are identical but have different sources. The set of hexamers identified from our method produced an overlap for 54 ESE sequences comprising 641 nucleotides, out of 738, yielding a coverage of 87%. Twenty-eight of these sequences were perfectly identified by the hexamers covering all the nucleotides. The ESS sequences were not recognized as well as the ESE ones. These results are shown in Table 5.6.

Rescue-ESE [18], Fas-ESS [56] and ESR [21] are computational methods that are specifically tailored to identifying exonic signals that impact a splicing event. Rescue-ESE identified candidate exonic splicing enhancers in vertebrate exons based on their statistical features. This method identified a set of 238 hexamers, which we refer to as RescueESE. Fas-ESS started with a set of experimentally identified exonic splicing silencer sequences of length 10. It computationally derived a set of 176 hexamers which we refer to as FasESS. ESR identified exonic splicing regulator sequences based on conservation of synonymous nucleotides. This set contains 285 hexamers, which were not necessarily divided into enhancer and silencer categories.

We refer to this set as AstESR. An additional method (Zhang and Chasin, [65,67]) compared bona fide exons with pseudo-exons in order to identify putative ESEs (PESEs) and putative ESSs (PESSs). The PESE set contains 2060 octamers and the PESS set contains 1018 octamers. There were 1701 unique hexamers in the PESE set, which we refer to as ChPESE, and there were 924 unique hexamers in the PESS set, which we refer to as ChPESS.

In order to be able to compare the FGA-generated features with the ESE hexamers identified by these methods, we looked at the different FGA sets of features that contained six consecutive position-specific nucleotides and were associated with the exonic regions. We looked at the feature sets generated for both acceptor and donor splice sites. We selected the features that belonged to the sequence interval 80 nucleotides downstream of annotated acceptor splice sites and 80 nucleotides upstream of annotated donor sites (bearing in mind that these intervals can contain some contribution from the adjacent intron that lies beyond the exon). Because FGA features were position-specific, for each set we computed the interval-feature patterns, thus obtaining a list of hexamers found in the exonic regions. We divided the features into positively weighted and negatively weighted sets denoted as $S-kMERn[p_1,p_2]+$ and $S-kMERn[p_1,p_2]-$, where $S \in \{A,D\}$ stands for acceptor and donor features respectively.

We computed the overlap between each FGA-generated set of hexamers and each of the four published sets of exonic regulatory sequences. We present the overlap for each pair of sets and the corresponding p-values in Table 5.7 and Table 5.8. The p-value shows the probability that a randomly selected set of hexamers,

containing as many hexamer features as found by the FGA algorithm, has an overlap equal to or greater than the value given in the Overlap column in these tables; this probability is calculated from the hypergeometric distribution. In Table 5.7 and Table 5.8, we have highlighted all the p-values less than 0.01 or greater than 0.99, indicating the significant relationship between the feature sets. All of these other sets have significant overlaps with our features, but the most significant are with ChPESE and ChPESS sets, perhaps because they were generated using methods similar to ours.

In order to address possible positional preferences for ESE elements we examined the distribution of short motifs corresponding to ESEs among our features. We observed a clear preference for exon sequences, but did not find a strong preference for a particular interval or position. For example, the GAAG tetramer is weighted positively throughout the exonic region, as illustrated in Figure 5.6A and Figure 5.6B. This signal was found in almost every position in the 80 nucleotide region and the weights of the respective features were very similar, so we cannot specify a region or interval of preference. The one exception was the immediate neighborhood of the donor site (position -4), which reflects splice-site consensus rather than exonic splicing enhancer signal. In contrast, GAAG was a negatively weighted feature in the intronic region.

We next asked whether those hexamers present in our set but not others have predictive value. As described above, many experimentally determined exonic enhancers (as reviewed by Zheng [68]) overlapped our features. While this was true of the other sets as well, even when those previously described motifs were excluded,

our features still accounted for some observations. Interestingly, many of these were examples of the A/C-rich motifs: CACACA, GCCCAA, TCAACA, CATTCA and CCTACA. Such A/C–rich elements have been described before [14] but have not been extensively characterized.

5.4 Summary

In Chapter 3, we showed that our FGA algorithm could be used to build accurate sequence classifiers. Here we have shown that the features generated by our algorithm for the purpose of discriminating between true and false splice sites correspond to functional splicing signals. Generated features included known features such as the branch-site consensus, acceptor splice-site consensus, pyrimidine tracts, coding potential and exon splicing regulator signals. The ability of FGA to accurately extract the branch-site signal (Tables 4.2-4.4) is especially noteworthy in view of the elusive nature of this signal [31]. Furthermore, the generated features provided information about the preferred location and sequence of these features, as illustrated by the distribution of branch-site and pyrimidine-tract features. However, we note that because FGA does not produce features to capture particular events such as AG di-nucleotide exclusion zones [20], it was not able to extract contingent signals such as distant branch sites coupled to them.

In addition, novel aspects of splicing signals could also be inferred from this method. We point to two examples. One is the co-occurrence of a peak of CCTT scores with a group of negative CMLS weights for TTTT at position -11 in the acceptor region. We believe that this may be a real, and previously unappreciated, aspect of the pyrimidine-tract signal. This signal is recognized by the large subunit of

U2AF (and by PUF60; [24]). We note that in-vitro selection experiments [49] found a marked preference for a CC dinucleotide in the case of U2AF but not PTB or Sxl. Thus, although U2AF will bind oligoU, there are other proteins that will do so and these are generally splicing repressors. Our observed features were consistent with the possibility that positions -12 and -11 may be an especially important region for discriminating between positive factors and negative factors that bind to similar sequence elements. This subtlety was revealed by our features despite the fact that it was not apparent from raw nucleotide-frequency data (Fig. 4.9). In a second example, even though our ESE hexamer features showed a statistically significant overlap with those obtained by other computational methods (Tables 4.5A and 4.5B), there were examples obtained by ours but not other methods, including a number of ESE motifs that corresponded to experimentally determined exonic splicing enhancers.

Finally, this method can be easily applied to other species and to similar classification problems for the discovery of species-specific regulatory elements. We have made our features available online (www.spliceport.org).

5.5 Tables of Chapter 5

Table 5.1. Individual classification performances of FGA-generated feature sets for acceptor and donor splice sites.

FGA-generated feature sets for splice sites and their individual performances at splice-site prediction. Each value reported is an average precision (positive predictive value, TP/(TP+FP)) over 11 values of recall (sensitivity, TP/(TP+FN)), 0%, 10%, 20% ... and 100%, and is the result of a three-fold cross validation. All the features in these features sets extend along the whole splice-site neighborhood [-82, 80] that we study.

A-3mer	86.46	A-4mer	84.92
A-3mer1	84.16	A-4mer1	77.28
A-3mer2	77.01	A-4mer2	69.10
A-3mer3	69.42	A-4mer3	63.11
A-3mer4	63.30	A-4mer4	56.66
A-3mer5	56.84	A-4mer5	49.23
A-3mer6	49.50	A-4mer6	41.02
A-3mer7	41.22		
A-5mer	80.60	A-6mer	68.64
A-5mer1	69.20	A-6mer1	61.72
A-5mer2	62.74	A-6mer2	54.65
A-5mer3	56.25	A-6mer3	47.19
A-5mer4	49.08	A-6mer4	39.62
A-5mer5	40.51		

D-3mer	86.79	D-4mer	85.21
D-3mer1	83.45	D-4mer1	81.14
D-3mer2	80.31	D-4mer2	70.47
D-3mer3	70.08	D-4mer3	55.38
D-3mer4	56.06	D-4mer4	44.77
D-3mer5	42.97		
D-5mer	83.64	D-6mer	75.03
D-5mer1	77.20	D-6mer1	66.68
D-5mer2	57.42	D-6mer2	43.31
D-5mer3	38.09		

Table 5.2. Top scoring features in branch site interval

The 20 top-scoring $A-3mer3[-40,-20]$ features (i.e. composite features that start in the interval between -40 and -25 derived using FGA from a seed of trimers) are all related to either the branch-site consensus or the pyrimidine tract.

FGA A-3mer3[-40,-20] features	Weight
----------ctgacc-------	0.1800
----------ctgacc--------	0.1678
----------ctgacc---	0.1488
----------ctgacc---------	0.1453
----------cctgac------	0.1417
----------cctgac----	0.1382
----------tgaccc---	0.1371
--------ctgacc----------	0.1370
----------cctgac--	0.1368
------ctgacc----------	0.1359
----------ctgacc-----	0.1358
----------tctctc	0.1303
----------ccttct-	0.1283
----------cttttc	0.1281
----------ctttt-	0.1281
----------ctcacc------	0.1254
----------ctcacc--------	0.1219
----------ctgact----	0.1206
----------cctgac--------	0.1202
----------tccctc	0.1200

Table 5.3. Identified interval-feature patterns in the branch-point interval

The first column shows the interval-feature patterns in the branch-point interval [-40,-20]. The second column shows the number of individual occurrences for each pattern in different positions within the specified interval. The average assigned weight is given in the third column. For comparison we include the total number of occurrences for this pattern in the complete neighbourhood ([-82, 80]) (forth column), and in the last column we show the narrowed range interval that comprises the total occurrences for each pattern.

A-3mer3 [-40,-20]*	Interval occurrences	Average Weight	Total occurrences	Total Range
--cctgac--	10	0.096	13	[-34,-16]
---ctgacc-	9	0.131	12	[-33,-16]
---ctgact-	8	0.082	11	[-32,-16]
-ccctga---	7	0.083	7	[-32,-19]
--gctgac--	7	0.083	8	[-34,-18]
--tctgac--	7	0.083	8	[-32,-18]
----tgaccc	6	0.089	9	[-32,-16]
--actgac--	5	0.059	6	[-33,-13]
---ctgatg-	5	0.068	7	[-36, 18]
-cccctc---	7	0.065	24	[-35, 0]
---cctctc-	5	0.049	22	[-36, 0]
--gggagg--	6	-0.041	23	[-34, 14]
--aaaaaa--	5	-0.028	84	[-50, 80]

Table 5.4. Individual position-specific GCTGAC features

A summary of position-specific GCTGAC features and their respective weight assigned by the CMLS classifier from the A-3mer3[-40,-20] feature set.

Features in exact position wrt AG consensus	Weight
----------gctgac----------AG	0.114
----------gctgac----------AG	0.114
----------gctgac----------AG	0.105
----------gctgac----------AG	0.082
----------gctgac----------AG	0.077
----------gctgac----------AG	0.074
----------gctgac----------AG	0.068
----------gctgac----------AG	0.062

Table 5.5. Weight distribution comparison for tetramers CTTT, TCTT, TTCT, and TTTC.

These features are a subset of $A-3mer1[-60,-5]$. Note that the distributions of scores correspond to the well-known pyrimidine tract with the additional information that C is preferred to T at positions -15 through -11.

-24	tttc----------	0.019	ttct----------	0.041	tctt----------	0.061	cttt----------	0.031
-23	-tttc----------	0.077	-ttct----------	0.035	-tctt----------	0.041	-cttt----------	0.045
-22	--tttc----------	0.060	--ttct----------	0.026	--tctt----------	0.079	--cttt----------	0.073
-21	---tttc----------	0.071	---ttct----------	0.041	---tctt----------	0.050	---cttt----------	0.095
-20	----tttc----------	0.092	----ttct----------	0.093	----tctt----------	0.103	----cttt----------	0.122
-19	-----tttc----------	0.088	-----ttct----------	0.054	-----tctt----------	0.085	-----cttt----------	0.093
-18	------tttc---------	0.090	------ttct---------	0.072	------tctt---------	0.125	------cttt---------	0.099
-17	-------tttc--------	0.083	-------ttct--------	0.109	-------tctt--------	0.114	-------cttt--------	0.111
-16	--------tttc-------	0.104	--------ttct-------	0.125	--------tctt-------	0.085	--------cttt-------	0.119
-15	---------tttc------	0.159	---------ttct------	0.110	---------tctt------	0.141	---------cttt------	0.152
-14	----------tttc-----	0.124	----------ttct-----	0.117	----------tctt-----	0.119	----------cttt-----	0.074
-13	-----------tttc----	0.121	-----------ttct----	0.154	-----------tctt----	0.149	-----------cttt----	0.008
-12	------------tttc---	0.055	------------ttct---	0.120	------------tctt---	0.157	------------cttt---	0.127
-11	-------------tttc--	0.106	-------------ttct--	0.062	-------------tctt--	0.140	-------------cttt--	0.085
-10	--------------tttc-	0.163	--------------ttct-	0.072	--------------tctt-	0.092	--------------cttt-	0.169
-9	---------------tttc	0.122	---------------ttct	0.078	---------------tctt	0.077	---------------cttt	0.076

Table 5.6

FGA-generated feature sets	Nr of features
D+: donor 6mer positively weighted	701
D-: donor 6mer negatively weighted	271
A+: Acceptor 6mer positively weighted	263
A-: Acceptor 6mer negatively weighted	202

Notation:

in blue: donor features, in red: acceptor features overlap that do not appear in donor features

in caps: positive features, in black: no overlap nucleotides, blue/red low case: negatively weighted

		Overlap				
ESE	Length	D+	D-	A+	A-	Total
GACGACGAG	9	9	0	0	0	9
GATGAAGAG	9	9	0	8	0	9
AAGAAGAAG	9	9	0	9	0	9
GAAGGA	6	6	0	0	0	6
GAAGAA	6	6	0	6	0	6
Gctgagt	7	0	6	0	6	6
gAGGAAGAGAAAAGGGCAGCAGAGGAGAGgca	32	28	6	12	6	31
GAAGAAGAAG	10	10	0	10	0	10
GCAGCACCTGGc	12	11	0	12	0	12
gAGGAAG	7	6	0	0	0	6
GGAAGAAGATAAAGac	16	14	0	9	0	14
CCAGAAGGAac	11	9	0	0	0	9
gAGGAAGgtg	10	6	0	0	0	6
AGAAAGAAGAAA	12	12	0	8	0	12
AAGAAGAGg	9	8	0	7	0	8
AAGAAGCgaa	10	7	0	6	0	7
AAGAAGAAAAAAGAAGAAA	19	19	0	18	0	19
gGTGACCTGCTGCAG	15	14	6	15	6	15
CTGCGGGACGATGTGCAGAG	20	20	0	6	0	20
GAAGAAGA	8	8	0	8	0	8
GAAGAAGAC	9	9	0	8	0	9
AAGAAGAAG	9	9	0	9	0	9
aagAGGACCCGCAGGC	16	13	0	8	0	13
AGGACAA	7	7	0	0	0	7
TGGACCCAGAGgt	13	11	6	7	0	11
GAAGAGGAAG	10	10	0	0	0	10
GAAGAA	6	6	0	6	0	6
Ggaagg	6	0	0	0	0	0
GAAGAAGCGGAGACAGCGACGAAGA	25	25	0	13	0	25
GAAGAAGAA	9	9	0	9	0	9
GGAGAAAGGAGAGa	14	13	0	0	0	13
GAGATGTGATGAAGGAGATGGgagg	25	21	9	13	7	25

Table 5.6 cont.

ATCCAGGAGGGGAACAGa	18	17	0	9	0	17
GAAGGACAGCA	11	11	0	0	0	11
AAGAAGGAa	9	8	0	6	0	8
AGAGATCGAGGAGGAtTTGAGAg...(22nt)...GAAGAAAGA	32	30	6	15	0	31
gggGGGAAGCACACAGAGCCCAACGAGACCAc	32	28	6	21	6	28
CAGACAa	7	6	0	0	0	6
AAGAAGGAAGg	11	10	0	6	0	10
GAAGAAGAA	9	9	0	9	0	9
agAGGAAGGCGA	12	10	0	0	0	10
AGGAGCAGgGGACGAAG	17	16	0	6	0	16
aAGAGAAG	8	7	0	6	0	7
GAGGAGGAG	9	9	0	9	0	9
GAGGAGGAG	9	9	0	9	0	9
GAGGAGGAG	9	9	0	9	0	9
GAAGAAGAG	9	9	0	8	0	9
GAAGAAGAG	9	9	0	8	0	9
ACCACCACC	9	9	0	7	0	9
ACTTCAACAAGtt	13	11	0	6	0	11
CAACCACAa	9	8	0	0	6	8
cacCATTCACGACACC	16	13	6	9	0	16
CAAGCATCAGCAAAAGCCAAac	22	20	0	6	0	20
Tgtcgattcca	11	0	0	0	0	0
Tgccgtt	7	0	0	0	0	0
Tgctgtt	7	0	0	0	0	0
tCCTACATCCT	11	10	0	0	0	10
Tgtcgattcca	11	0	0	0	0	0
Tgccgtt	7	0	0	0	0	0
Tgctgtt	7	0	0	0	0	0
tCCTACATCCT	11	10	0	0	0	10
Total:	**738**	**622**	**51**	**356**	**37**	**641**

Table 5.6 cont.

ESS	Length	Overlap D+	Overlap D-	Overlap A+	Overlap A-	Total
Tttgaa	6	0	0	0	0	0
Tcttctt	7	0	6	0	0	6
Ggctccccc	9	0	0	0	0	0
AGAGCAGg	8	7	0	0	0	7
Tggt	4	0	0	0	0	0
ctagaTATGGATCC	14	7	0	6	0	9
GTGACCCCttacctaCTCACACCACtgcATTCTCaccgc	40	24	17	6	17	32
AAGCACcttg	11	6	6	6	0	8
ccaAGTCAAaatttac	16	6	7	0	8	11
Tag	4	0	0	0	0	0
Tctttaggttccctttcaattct	23	0	12	0	0	12
CAAGGCc	7	6	0	0	0	6
Catgg	5	0	0	0	0	0
Ctagactaga	10	0	0	0	0	0
Ttgggt	6	0	0	0	0	0
Ttag	4	0	0	0	0	0
Pytag	5	0	0	0	0	0
ccaatagtagtagcgGGAGAAtg	23	6	16	12	10	18
ctagtaaacttattcttacgtctttcctgtgttgcCCTCCAGCTtttatctctGAGATGGtcttctttctaga	73	16	40	0	24	44
Agttcca	7	0	0	0	0	0
ttaAACACAAGtt	13	8	0	0	0	8
Tagaca	6	0	0	0	0	0
Taagtgttctgagct	15	0	6	0	0	6
tgtggGGGACC	11	6	0	0	0	6
Total	**327**	**92**	**110**	**30**	**59**	**173**

Table 5.7. FGA-generated feature set show significant overlap with ESE regulator signal sets.

The number of shared features between the FGA generated sets of hexamers and the AstESR, RescueESE and PESE hexamer sets and the p-value stating the probability of having this overlap or a greater overlap by chance. We highlight the highly statistically significant probabilities. The set $D-3mer3[-80,-1]$ did not contain position specific hexamers and the set $D-4mer2[-80,-1]$ contained only 3 position specific hexamers, two of which overlapped with RescueESE set.

FGAset	size	AstESR (285) Overlap	AstESR (285) P-value	RescueESE (238) Overlap	RescueESE (238) P-value	ChPESE (1701) Overlap	ChPESE (1701) P-value
A-3mer3[1,80]	313	34	0.00514	24	0.09415	175	2.09e-06
A-3mer3[1,80]+	177	28	0.00003	24	0.00007	130	1.42e-18
A-3mer3[1,80]–	136	6	0.92089	0	*	43	0.9939
A-4mer2[1,80]	317	35	0.00347	26	0.04319	177	1.96e-06
A-4mer2[1,80]+	179	29	0.00001	25	0.00003	129	2.74e-17
A-4mer2[1,80]–	138	6	0.92714	1	0.99999	46	0.9819
A-5mer1[1,80]	342	35	0.01147	27	0.05920	278	1.06e-08
A-5mer1[1,80]+	187	29	0.00003	25	0.00006	134	1.40e-17
A-5mer1[1,80]–	155	6	0.96496	2	0.99915	59	0.8352
A-6mer[1,80]	465	54	0.00006	27	0.53401	278	1.06e-08
A-6mer[1,80]+	263	38	0.00001	25	0.00899	165	6.61e-13
A-6mer[1,80]–	202	16	0.32994	2	0.99984	76	0.8907
D-5mer1[-80,-1]	64	10	0.01195	32	1.32e-23	60	5.59e-19
D-5mer1[-80,-1]+	56	9	0.01403	30	2.47e-23	52	4.27e-16
D-6mer[-80,-1]	1052	126	1.44e-12	112	1.81e-13	613	3.73e-37
D-6mer[-80,-1]+	701	93	2.28e-11	109	6.16e-28	482	1.02e-57
D-6mer[-80,-1]–	271	20	0.42504	1	0.99999	90	0.9985

* p-value is very close to 1.

Table 5.8. FGA-generated feature set overlap with ESS regulator signal sets.

The number of shared features between the FGA generated sets of hexamers and the FasESS and PESS hexamer sets and the p-value stating the probability of having this overlap or a greater overlap by chance. We highlight the highly statistically significant probabilities.

FGAset	size	FasESS (176) Overlap	FasESS (176) P-value	ChPESS(924) Overlap	ChPESS(924) P-value
A-3mer3[1,80]	313	10	0.877	73	0.5407
A-3mer3[1,80]+	177	1	0.999	8	*
A-3mer3[1,80]–	136	9	0.129	59	3.19e-08
A-4mer2[1,80]	317	10	0.887	72	0.6423
A-4mer2[1,80]+	179	1	0.999	9	*
A-4mer2[1,80]–	138	9	0.137	57	4.22e-07
A-5mer1[1,80]	342	12	0.812	70	0.9300
A-5mer1[1,80]+	187	3	0.999	9	*
A-5mer1[1,80]–	155	9	0.221	54	0.000257
A-6mer[1,80]	465	17	0.799	91	0.9993
A-6mer[1,80]+	263	7	0.943	19	*
A-6mer[1,80]–	202	10	0.368	64	0.001374
D-5mer1[-80,-1]	64	1	0.941	4	0.9999
D-5mer1[-80,-1]+	56	0	*	4	0.9995
D-6mer[-80,-1]	1052	26	0.999	183	0.9999
D-6mer[-80,-1]+	701	6	0.999	63	*
D-6mer[-80,-1]–	271	19	0.022	106	1.54e-10

* p-value is very close to 1.

5.6 Figures of Chapter 5

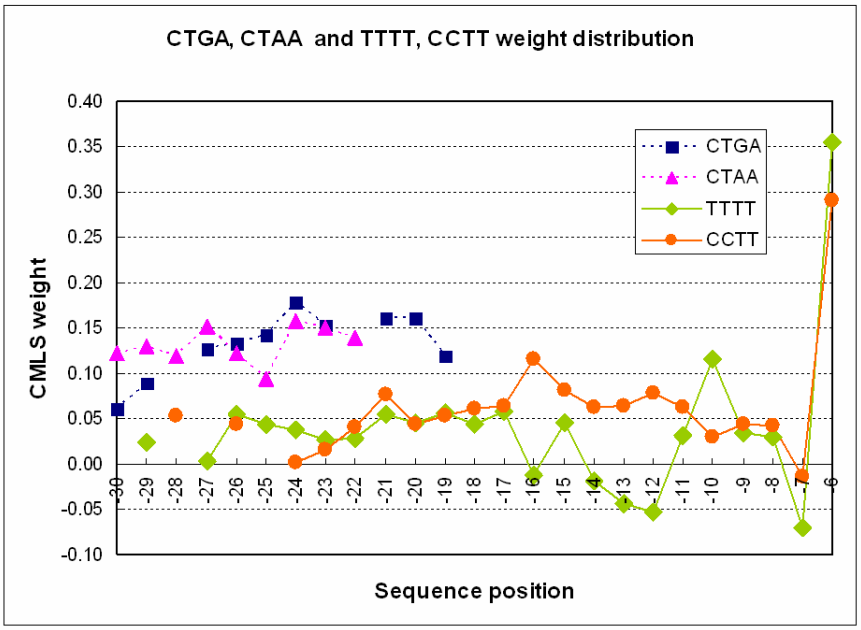


Figure 5.1. Weight distribution comparison for pairs of tetramers CTGA, CTAA and TTTT, CCTT.

The distribution of CMLS weights for four tetramers from $A-3mer1[-60,-5]$ is shown graphically. Note that the distributions of scores for CTGA and CTAA are similar and sharply focused around the peak that would place the branch A at position -24. Note that the distributions of TTTT and CCTT corresponds to the well-known pyrimidine tract with the additional information that C is preferred to T at positions -15 through -11, where a peak of scores for CCTT coincides with a group of negative values for TTTT. There are no occurrences of these four hexamers in this feature set upstream of the region shown.

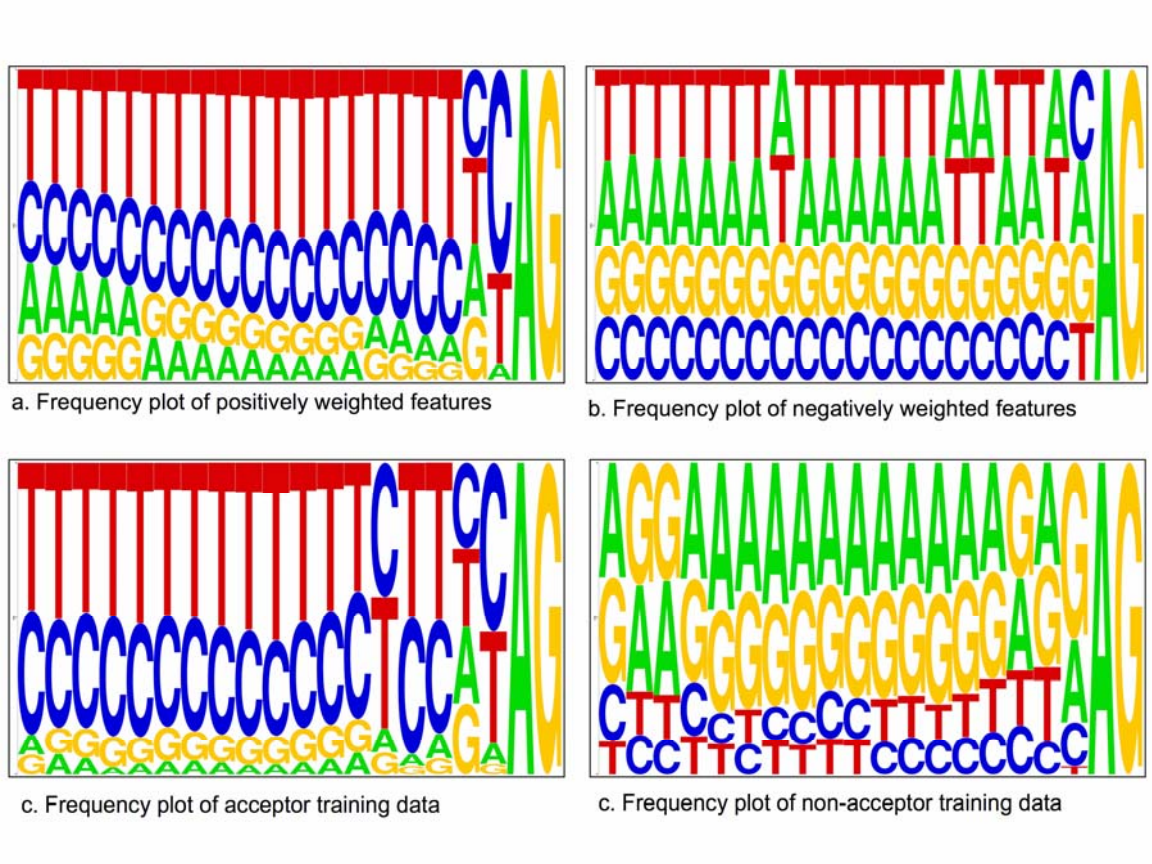


Figure 5.2. The acceptor splice-site (pyrimidine-tract) interval.

Frequency plot sequence logos for the positively and negatively weighted features in the pyrimidine-tract interval, $A-5mer[-20,-1]$, (Figure 5.2a and Figure 5.2b), compared with frequency distribution of the training acceptor and non-acceptor sequences in the same interval (Figure 5.2c and Figure 5.2d). The positive features frequency plot corresponds to the acceptor splice-site consensus, which is also illustrated with the true acceptor sequences frequency plot. The negative features frequency plot reveals an AG-rich element.

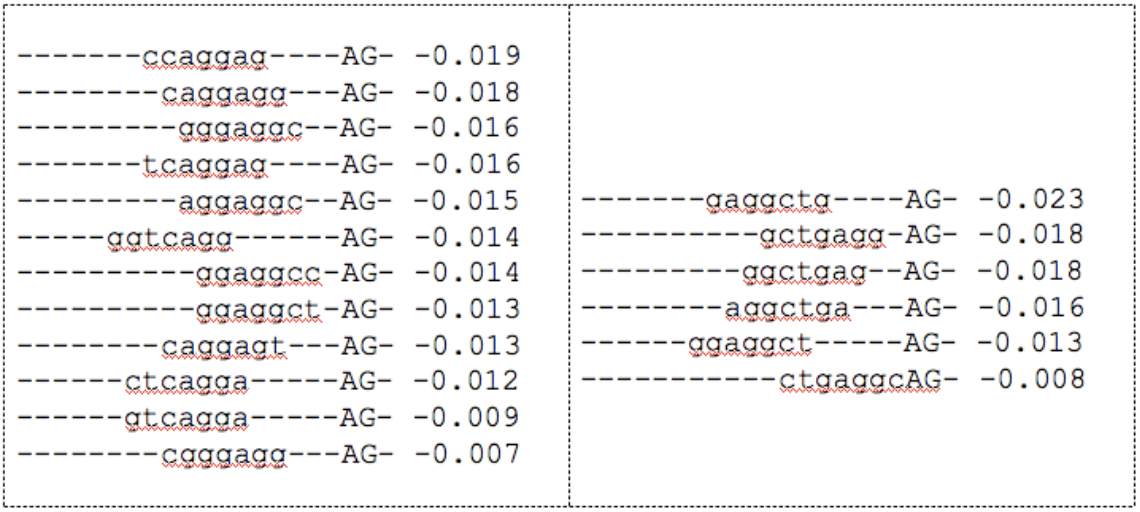


Figure 5.3. Clusters of negative features of the pyrimidine-tract interval.

Examples of the individual features for two clusters of features and the assigned CMLS weight for each feature from the feature set $A-6mer1[-20,-1]$. The presence of the AG dinucleotide upstream the annotated 3' splice site, in the pyrimidine-tract interval is not preferred. All these features have negative weights.

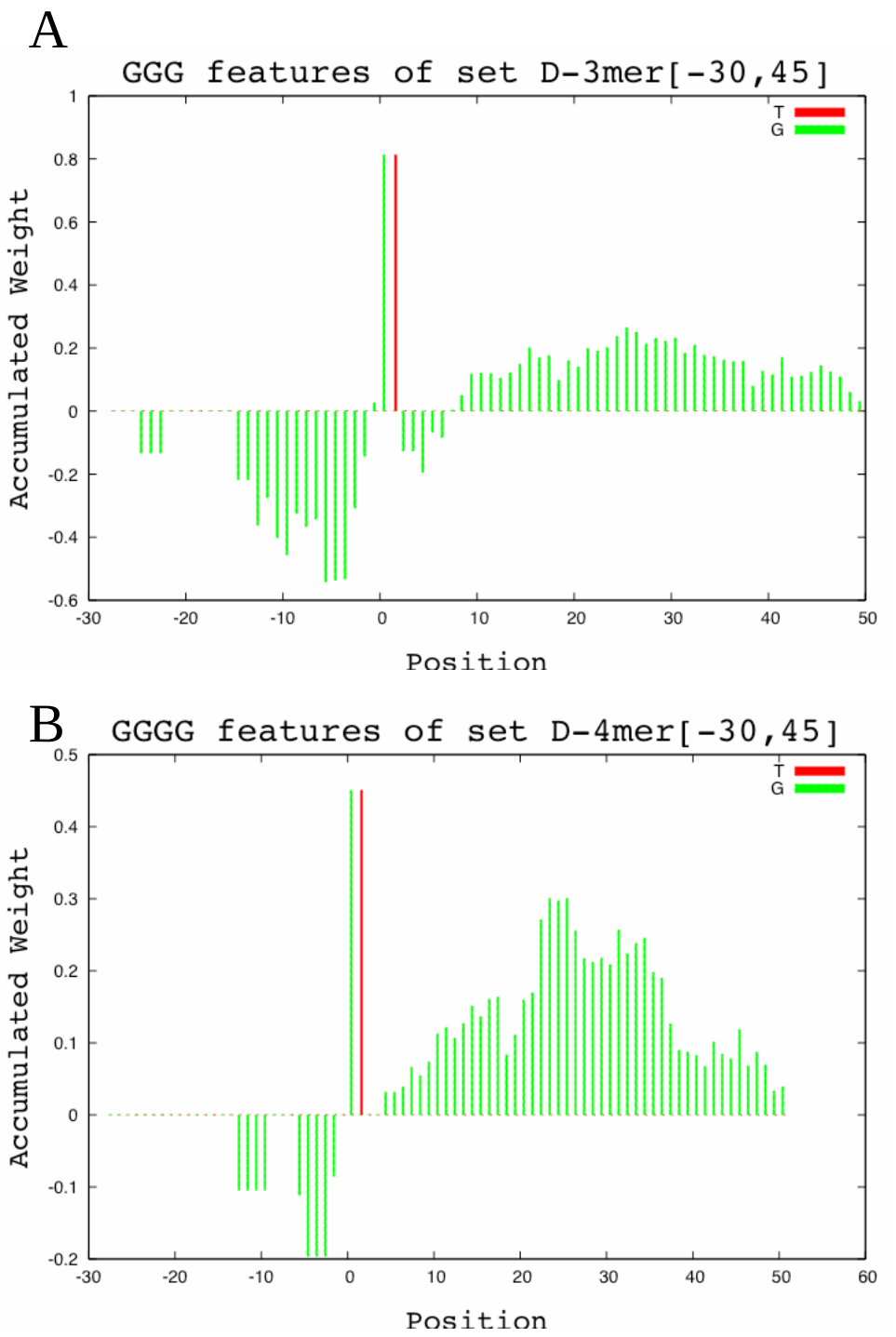


Figure 5.4. G-rich features in the donor-site interval.

Weighted histogram for all the GGG and GGGG features in the donor-site interval selected from the feature sets $D-3mer3[-30,45]$ and $D-4mer2[-30,45]$. These features are not preferred upstream the donor site, but they are encouraged on the downstream region.

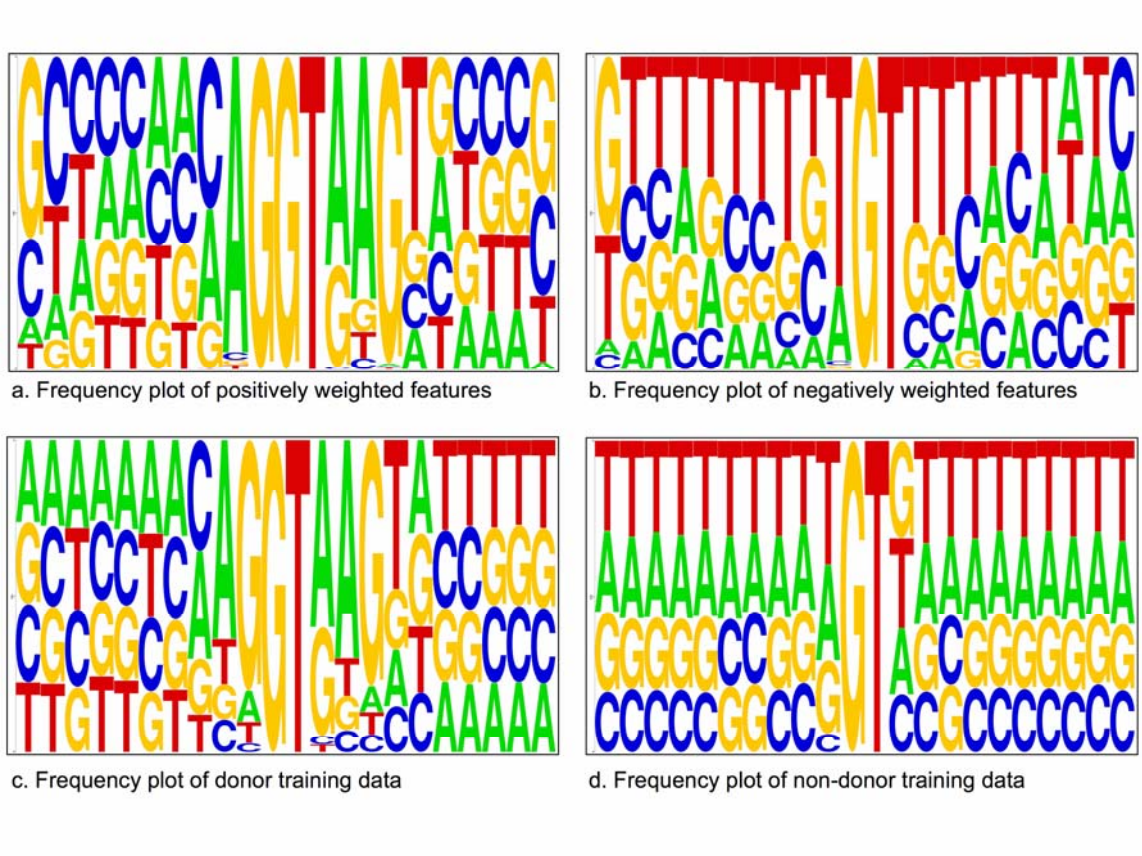


Figure 5.5. The donor splice-site interval.

Frequency plot sequence logos for the positively and negatively weighted features in the donor-site interval, $D-6mer[-10,10]$ (parts a and b), compared with frequency distribution of the training donor and non-donor sequences in the same interval (parts c and d). The positively weighted features capture the donor-site consensus ([A|C]AGGT[A|G]AGT.

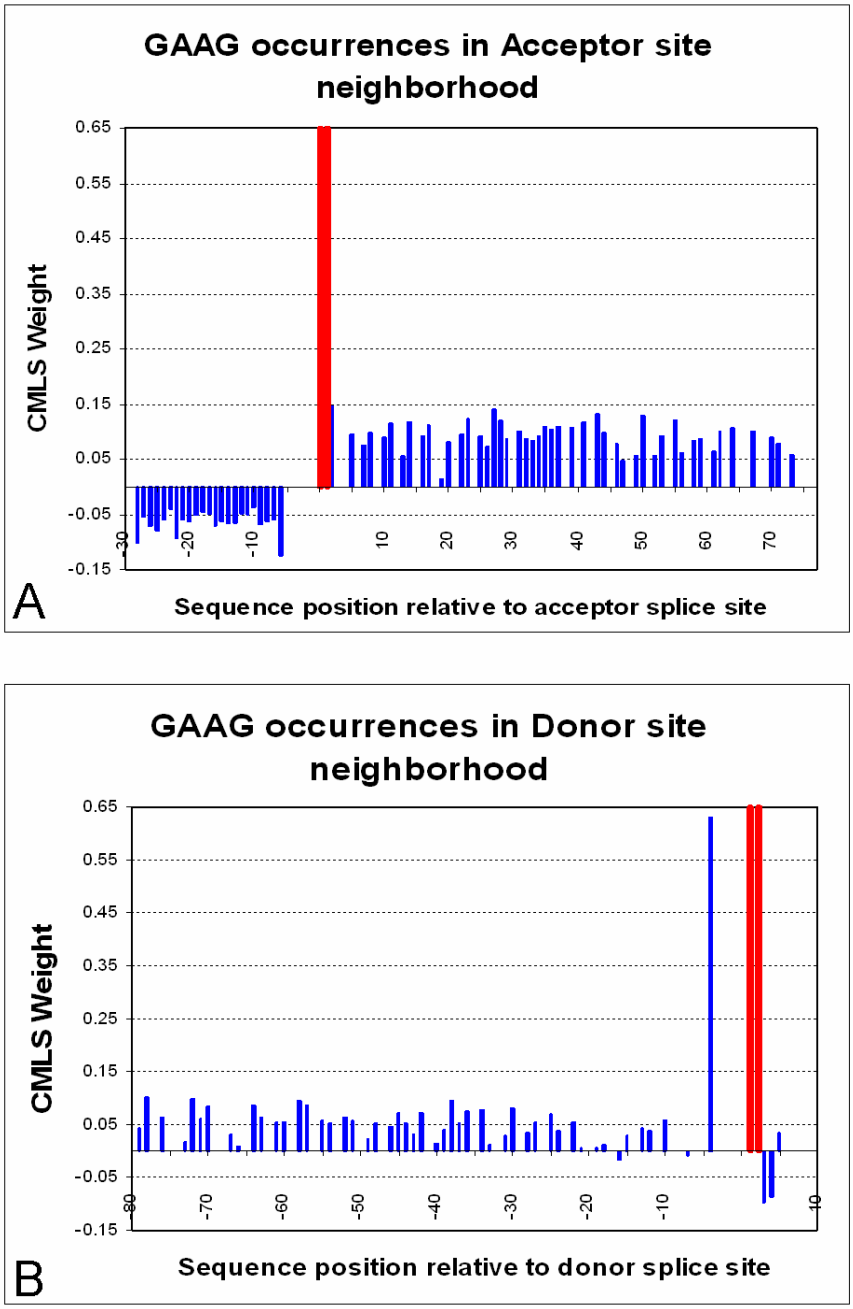


Figure 5.6. The weight distribution of the ESE motif GAAG in the donor splice-site neighborhood.

The x-axis shows the splice-site neighborhood interval. The consensus dinucleotides AG and GT locations are marked with the red bars (positions -2,-1, in A and positions 1, 2 in B). For every

occurrence of the feature GAAG in the sets $A-4mer[-80,80]$ (Figure 5.6A) and $D-4mer[-80,80]$ (Figure 5.6B), we draw a bar corresponding in height to its CMLS assigned weight. This feature has a negative weight when it is positioned in the intronic region, but a positive weight in the exonic region. We notice its exceptionally high weight at position -4 in Figure 5.6B. One possible reason may be the reflection of the donor-site consensus signal.

Chapter 6: Generating RNA secondary-structure features

RNA molecules are distinguished by their sequence composition and by their three-dimensional shape, called the secondary structure. The secondary structure of a pre-mRNA sequence may have a strong influence on gene splicing. In Chapter 3, we showed that a splice-site model employing sequence features built by using our feature-generation algorithm was very effective in predicting splice sites. The generated sequence features also contained biologically relevant features, as described in Chapter 5. In this Chapter, we extend the feature-generation algorithm to construct secondary-structure features. These features capture the nucleotide-pairing tendency in the splice-site neighborhood. We extend the splice-site model to include both pre-mRNA sequence and structure characteristics. The new model outperforms the sequence-based features model. The identified secondary-structure features capture biologically relevant signals, such as splicing silencers. We also find that these signals are concentrated in specific regions around the splice-site neighborhood and we detail their characteristics.

6.1 Secondary structure of nucleic-acid sequences

The secondary structure of RNA molecules is defined by the pairings of the nucleotides along the sequence. RNA secondary-structure characteristics are important in biology because RNA sequences fold into structures that are critical to

their biological functions. Secondary-structure properties may also help identify subsequences of nucleotides that interact with other molecules or complexes.

Human genes — and the genes of every eukaryotic organism — are composed of contiguous coding regions in the DNA sequence. Non-coding regions, introns, separate the coding regions, exons. Messenger RNA copies the portion of the DNA that contains a gene (pre-mRNA), and during the splicing process, the non-coding regions are excised from the pre-mRNA sequence. All the coding pieces, then, are ligated together into the final gene product (mRNA), ready to be translated into protein. Splicing takes place in several stages. There are a number of proteins that can recognize splice-site locations and bind to the sequence, facilitating the intron excision.

Splice-site prediction is the task of recognizing the actual boundaries of the protein-coding regions in the DNA sequence. Accurate splice-site prediction is a critical component of gene prediction. Gene prediction from DNA sequence data is an important goal in bioinformatics, not only to provide fast and reliable annotation of the large quantity of sequences data, but also to provide valuable biological insights. In Chapter 3, we developed a splice-site prediction model achieving significant accuracy improvements over existing methods. In Chapter 4 and Chapter 5, we showed that the features generated using FGA correspond to biologically significant functional elements.

So far, in our splice-site prediction model, we have considered only sequence-based features. However, the splicing process is not a mere linear process. In fact, the correct identification of the splicing borders actually involves a large number of

proteins. The affinity of sequence nucleotides to form pairing bonds may guide these proteins to their binding sites, thus having an important effect in the splicing process. To investigate this, we use a very effective RNA secondary-structure prediction algorithm [43] to fold the training sequences into their secondary-structure form. Using the secondary-structure sequences, we extend our feature generation algorithm to generate structure-based features.

The combined splice-site model of both sequence- and structure-based features improves splice-site prediction. The secondary-structure features also capture important biological properties.

The possibility of extracting useful information from RNA secondary structure for splice-site prediction was proposed by Patterson et al. in [44]. Their splice-site prediction model combined a sequence-based splice-site predictor score and a few structure-based metrics, such as the optimal folding-energy score, the max-helix score, and a second-order Markov model to capture the pairing profile of a folded sequence. They suggested that there are structural cues that should be exploited by gene-finding algorithms.

Our approach differs from [44], in that we searched the space of possible position-specific nucleotide pairings in order to find specific features that improved splice-site prediction. We also offer biological interpretation for the identified features. Our recent work demonstrated that our sequence-based splice-site predictor achieved significantly better results than the WAM model, which was used as the sequence-based predictor in their work.

This chapter is organized as follows. Initially, we describe our data and their secondary-structure form. Next, we describe how we expand the feature-generation algorithm to generate structure-based features. Here, we also give a brief summary of the definitions of the sequence-based features used in the splice-site prediction model. Next, we provide a detailed description of our experiments, using the secondary-structure features. Finally, we discuss our findings and the possible biological relevance of the new features, and we conclude with several future directions.

6.2 Data characteristics

The dataset used for feature generation was the same collection of 162-nucleotide-long training sequences centered at the splice site, as described in Chapter 3. Both upstream and downstream regions were 80 nucleotides long and the sequence alphabet was {A,C,G,T}. The acceptor-site training data contained 20,996 positive instances and 200,000 negative instances, and the donor-site training data contained 20,761 true positive instances and 200,000 negative instances.

We used these sequences to generate sequence-based features, as described in Chapter 3. For secondary structure characteristics, we need the three-dimensional shape. We used the RNA secondary-structure prediction algorithm, Afold [43], to fold all the training sequences into their three-dimensional form. Alexey Ogurtsov, the author of Afold, modified the output of Afold such that, for each input sequence, it produced as output the additional information of the nucleotides which were paired in the secondary structure. Those constituted the secondary-structure sequences, and an example is shown in Figure 6.1. In this figure, we see two sequences termed S1 and S2. These sequences have exactly 162 nucleotides, some of which are shown in

uppercase and some in lowercase characters. The nucleotides which are shown in uppercase are paired in the secondary structure. Given this representation, we can think of other features to consider, such as position-specific k-mers that may participate in pairing bonds in the three-dimensional form of the given sequence.

This method has several disadvantages. First, the secondary-structure information for each sequence is computed by a computational method (Afold). As a result, what we are taking as ground truth, may not be correct, because we have not, cannot, validate these folding predictions. It is, however, claimed by the authors that Afold is very accurate and efficient in producing the secondary structure of pre-mRNA sequences. Second, we are considering only the first best result of Afold for each sequence. A more prudent approach might be to consider the top ten structure predictions of Afold for each sequence. We preferred to choose one prediction for each sequence, since in this way we could compute features for our training dataset in an absolute manner, by distinguishing among features that better separated the two classes of sequences. If we considered many possibilities, then we would have to attach a probability value for each feature and this is out of the scope of this thesis. However, it would be an interesting future direction.

So, given a single secondary-structure folding per sequence, we wanted to determine whether splicing was affected by the pairing tendency of the nucleotides in the close neighborhood of the splice site. To pursue that question, we plotted the fraction of positive sequences having a paired *k*-nucleotide subsequence (*k*-mer) for each position of its length and compared it with that of the negative sequences. Those plots are shown in Figure 6. and Figure 6.. We were surprised to see that for acceptor

splice-site sequences, the positive sequences showed a higher tendency to have paired *k*-mer sequences in the upstream region, with a clear peak of pairing tendency just before the actual splice-site position. The donor splice-site sequences, on the other hand, showed a tendency toward reduced *k*-mer pairings in the upstream region and a higher tendency for pairing in the downstream region.

These observations are of special interest because they are consistent with the actual splicing scenario that takes place in living cells. These findings encouraged us to investigate the possible impact of secondary-structure features on splice-site prediction.

6.3 Feature Generation for Splice-Site Prediction

This section summarizes our feature-generation algorithm (FGA) and describes the new feature-construction procedures for the generation of secondary-structure features. Recall that FGA uses domain knowledge and data properties to construct and select useful features for the prediction task. Starting with an initial feature set, FGA iteratively calls a feature-construction method to expand the current feature set and a feature-selection method to reduce the feature set size to manageable levels. After a specified number of iterations, the algorithm produces an output feature set. Those features are, in turn, used by a classification algorithm for the classification task. We consistently used the classifier CMLS in our experiments because of its superior performance in comparison with other classification algorithms, such as: AdaBoost, Naïve Bayes, logistic regression, and support vector machines.

6.3.1 Feature Construction for Splice-site prediction

The first stage of the feature generation algorithm generates feature sets useful for splice-site prediction. Initially, we define the basic elements to construct features. In the case of pre-mRNA sequences, we use the nucleotide alphabet and sequence length to construct sequence-based features.

Feature construction for sequences: As described in detail in Chapter 3, we considered several feature types that capture compositional and positional properties of sequences: general *k*-mer, upstream/downstream *k*-mer, position-specific *k*-mer, and conjunctive positional features. We described these features and their individual construction methods in Chapter 3. Here, we extend our algorithm to capture the secondary-structure characteristics of the splice-site sequence.

Feature construction for secondary-structure sequences: We define a novel feature type that captures the structure characteristics of the RNA sequences, the position-specific paired *k*-mers. A position-specific paired *k*-mer is a string of k nucleotides that, in the output sequence of the RNA secondary-structure algorithm, is predicted to form pairing bonds with other nucleotides in the sequence. To identify possible binding motifs for the proteins that affect splicing, we use our feature-generation algorithm to identify useful position-specific paired *k*-mer features.

The position-specific paired k-mers are Boolean features; for each sequence we record whether they are present or not. Given a sequence of length n, for each value of k, there are $(n-k+1)\times 4^k$ position-specific paired k-mers.

Construction Method: This construction method starts with an initial set of position-specific paired *k*-mer features and expands them to a set of position-specific paired (k+1)-mers by appending letters of the alphabet to each feature. As an example, assume $F_{initial}$ is $\{A_1A_2C_3C_4\}$. This set contains one feature, the 4-mer "AACC" starting at the first sequence position. Each nucleotide of this feature is showed in capital letters, symbolizing the pairing in the secondary structure. Now, we can extend it to the next level set of position-specific paired 5-mers, $F_{constructed}$={ $A_1A_2C_3C_4A_5$, $A_1A_2C_3C_4C_5$, $A_1A_2C_3C_4G_5$, $A_1A_2C_3C_4T_5$}. The constructed feature set contains four 5-mers such that every nucleotide is paired in the secondary structure description. In that manner, we incrementally construct higher levels.

6.3.2 Feature Selection for Constructed Features

Feature selection: To reduce the size of our constructed feature sets, we considered different feature-selection methods: IG, CHI, MI, and KL. The definitions of these measures were described in Chapter 3.

Feature generation: For each initial feature set, we iterate between a feature-construction method (to obtain more complex features) and a feature-selection method (to reduce the dimensionality of the constructed set). We perform this process for a predefined number of iterations. In this manner, we generate different feature sets, each useful for splice-site prediction.

Recursive Feature Elimination: After we generate the individual feature sets separately, we collect all the features into a mixed set. Starting with the mixed set, we learn a prediction model using the CMLS classifier. CMLS produces a decision boundary that discriminates between the two different categories. Each feature is

assigned a weight during learning. These weights define the decision boundary and can be used for ranking. Features with zero weights, or weights very close to zero, are assumed to not contribute to the classification task, and are therefore eliminated. In this manner, we learn a new model and, after each iteration, eliminate a fixed number of features.

6.3.3 Splice-Site Prediction Model

Our generated features are of two major types: features capturing sequence properties and features capturing structure properties of the splice-site neighborhood. Using this natural separation, we use a classifier to learn sequence- and structure-features splice-site prediction models. Then, we define a new model for splice-site prediction — a linear combination of the structure-features model and the sequence-features model:

$$Score_{seq} = c_0 + c_1 \times Score_{structure} + c_2 \times Score_{sequence}$$

The structure-model and sequence-model of splice-site prediction are used to score a held-out training-sequences set. Then, we use the classifier to learn the coefficients for the linear combination of the models. In the next section, we give a detailed analysis of all the mentioned methods and their results for the problem of splice-site prediction.

6.4 Experiments and Discussion

In Chapter 3 we discussed the feature generation procedure for the sequence-based features, which we divided into compositional, positional and composite

positional features. Here, we discuss the feature generation procedure for features capturing information about the secondary structure of the splice-site neighborhood. The following experiments discuss the splice-site prediction effect of the nucleotides showing high pairing potential, the position-specific paired k-mer features. All the reported 11ptAvg precision values are the results of three-fold cross validations.

6.4.1 Position-specific paired *k*-mers

Similar to our position-specific sequence-based *k*-mer features, we constructed all the position-specific *k*-mers for k values ranging from 1 to 5. Analogous to the feature generation of sequence-based position-specific k-mer features, described in section 3.7.2, using the FGA in the uncoupled mode, we first note the splice-site prediction accuracy when using the complete sets of position-specific paired k-mer features, for k values from 2 to 5. Then, we scored the features, using the feature-selection methods, picked the top scoring 1000 features for each selection method, and used the top 1000 features to predict splice sites for both donor and acceptor sites. The results are shown in Table 6. 1.

Similar to sequence-based position-specific k-mer features, we find that both IG and Chi feature selection methods, select position-specific paired features that result in comparable accuracy results. We collected 4000 features from position-specific paired *k*-mer sets for k from 2 to 5. To this set, we added position-specific paired 1-mer features (648 for a 162 nucleotide-long sequence). We applied recursive feature elimination on those sets of features, as shown in Table 6. 2. Compared with individual results of our sequence-based features, the 11ptAvg precision performance of the position-specific paired *k*-mers was very promising. It clearly showed that such

a feature carried an important amount of information, which could possibly contribute to further understanding of the splicing mechanism.

6.4.2 Splice-site prediction with sequence- and structure-based features

In order to compute the effects of the generated structure-based features on splice-site prediction, we combined them with the sequence-based features generated in Chapter 3, in a mixed features model. Starting with Table 6.2, we selected a set of features from the position-specific paired *k*-mers to combine with our previously identified acceptor and donor sequence-features sets. We decided to include 3100 structure-based features. The mixed model for donor-site prediction contained a collection of 1675 sequence features and 3100 structure features. These models produced the following 11ptAvg precision results: 89.74% for acceptor splice sites and 89.46% for donor splice sites. Although producing a low rate of false positives and ranking well, these initial results did not produce better predictions, compared with our sequence-based feature model (see Chapter 3 for a comprehensive description of those results).

To understand the importance of the secondary-structure features for splice-site prediction, we conducted the following experiments. Starting with the whole set of sequence and structure features, we applied recursive feature elimination, eliminating 200 features for each iteration. Table 6. 3 shows a summary of the splice-site prediction results for both acceptor and donor datasets in our experiments. For each iteration, we list the number of features in the mixed-features model, as well as the 11ptAVG precision of the three-fold cross validation for splice-site prediction (acceptor and donor). For each case, we also listed the number of features that

described sequence composition and structure characteristics for each mixed feature set. Then, we picked out the sequence-based features and the structure-based features separately and trained the CMLS classifier. We built prediction models for each separate sequence- and structure-feature set and Table 6.3 lists also these individual 11ptAvg precisions.

From the results of these experiments, as shown in Table 6. 3, we made several observations. First, the sequence composition was of primary importance in defining a splice site. The 11ptAvg results of models built only on sequence features consistently showed high values. Second, specific nucleotide pairings of particular locations could be the key to the discovery of important binding sites. The 11ptAvg results of models built only on structure features were several times higher than random (10%). Third, secondary-structure information improves splice-site prediction, in addition to sequence-based features. For example, as shown in Table 6. 3, when the number of features was reduced to 3000 for the acceptor site problem, the addition of paired position-specific features increased the 11ptAvg from 89.69%, which was the result of sequence-based features, to 90.36%. This result was statistically significant with an alpha of 0.005.

6.4.3 New prediction model with sequence- and structure-based information

The results in Table 6. 3 suggests that adding structure-based features with the large mix of features does not produce a visible difference in splice-site prediction results. Instead, in order to profit from the information encoded in the newly generated features, we used the combined model. The combined model initially learns

two different splice-site models: one based on the structure features and one based on the sequence features. To illustrate this, we selected the feature set of size 3000 in Table 5.3. This set contained 1679 position-specific paired *k*-mers (structure features) and 1321 general, upstream, downstream, and position-specific *k*-mers and conjunctive positional features (sequence features). The 11ptAvg result for splice-site prediction of the structure-based features model was 60.42% and the 11ptAvg of the sequence-based features model was 90.19%.

We learned the new splice-site prediction model as a linear combination of the structure-features model and the sequence-features model. We trained the classifier and learned the coefficients that defined the linear combination model. The linear combination model produced an 11ptAvg precision of 91.46% for donor splice-site prediction. This result was an improvement over the 90.36% obtained when using the whole set of 3000 donor features (mixed), and over the 90.19% obtained when using only the sequence features, as shown in Table 5.3. This improvement is statistically significant for an alpha of 0.005.

6.5 Biological significance

As discussed in Chapter 4, the biological signals that are present in the splice-site neighborhood fall into these categories. Exonic splicing enhancers are signals that activate the nearby splicing sites. Exonic splicing silencers act as suppressors to the splicing activity. Both enhancing and silencing effects are accomplished via the different types of proteins that bind to the ESE and ESS signals. Fairbrother et al. [18] identified 238 candidate ESE 6-mers, the RescueESE set. Goren et al. [21] identified

a set of 285 candidate splicing regulator 6-mers, the ESR set. And Wang et al. [56] derived a set of 176 candidate ESS 6-mers, the FasESS set.

Because the FGA-generated secondary-structure features captured the pairing information of different nucleotides and their preferred locations, we hypothesize that these specific paired features may have discovered ESE and ESS sites in the splice-site neighborhood. To test that, we compared them with the published ESE and ESS sets. To compare our position-specific paired 5-mers with the exonic splicing regulator sets, we derived all the 5-mers contained in those 6-mers. The RescueESE set contained 208, the ESR set contained 297, and the FasESS set contained 142 unique 5-mers. We computed the overlap between our features and the 5-mers in the published regulator sets. For each overlap, we computed the p-value, based on the hypergeometric distribution.

The set of FGA-generated 5-mers of the downstream donor region produced a significant overlap with the FasESS set of splicing silencer signals (p-value=6.93e-19). The splicing silencer signals are more subtle signals and therefore more difficult to discover. The upstream donor region 5-mers produced a significant overlap with the ESR set of splicing regulator signals (p-value=5.04e-07).

To investigate these signals further, we selected the 5-mer features that produced the overlap, and we searched their exact positions in the splice-site neighborhood. We divided the neighborhood into six regions: the far, near, and close regions upstream and downstream from the annotated splice-site position. The far region upstream or downstream denoted the interval 50-80 nucleotides away from the splice site. The near region denoted the interval from 20 to 50 nucleotides and the

close region denoted the 20 nucleotides upstream or downstream the splice site. We grouped the overlapped 5-mer features into these six regions and we listed them in Table 6. 4. This detailed description has not been done before and we believe it will be of value to biologists. Although some of the signals appear in more than one region, it is interesting to note that the weight of the features also changed with their position, sometimes even switching sign.

6.6 Summary

In this Chapter, we presented an extension to our feature-generation algorithm, constructing features that capture the three-dimensional characteristics of genomic sequences. This algorithm was applied to the problem of splice-site prediction, and a new splice-site predictor model was proposed. The new model employed features that captured both sequence composition and structural-shape characteristics of splice-site sequences. The linear combination of a structure-features model with a sequence-features model improved splice-site prediction accuracy significantly. Moreover, the features employed by the structure-based model were found to overlap significantly with splicing regulator motifs. We divided the 160-nucleotide splice-site neighborhood into six regions, and we mapped the position preference of the identified biologically relevant signals. This detailed description may be valuable to biologists. In our future work, we plan to investigate other biologically relevant information, such as the identification and location of features that capture the tendency not to create a pairing bond.

6.7 Tables of Chapter 6

Table 6. 1

Feature-generation for position-specific paired *k*-mer features for k from 2 to 5 for acceptor and donor splice-site prediction. We give the 11ptAvg precision results for all the features and when top-1000 features are selected, using different methods.

Acceptor-site Models

K-mer	All	IG	KL	MI	Chi
1	61.79				
2	64.46	62.11	61.84	46.62	62.13
3	59.82	55.05	-	43.46	54.96
4	51.04	42.93	36.98	40.17	43.02
5	44.13	38.72	27.17	37.20	-

Donor-site Models

K-mer	All	IG	KL	MI	Chi
1	61.07				
2	66.08	61.88	61.78	44.29	61.92
3	-	54.73	53.09	47.91	54.61
4	51.21	44.06	41.30	39.42	43.40
5	45.29	43.15	35.12	41.37	43.70

Table 6. 2

Splice-site prediction results for position-specific paired *k*-mer features for different stages of recursive feature elimination, using CMLS. For each iteration we reduce the number of features by 500 and report the 11ptAvg for splice-site prediction.

Nr of Features	11ptAvg (Acceptor)	11ptAvg(Donor)
4600	66.81	69.77
4100	66.84	69.82
3600	66.91	69.17
3100	66.74	69.03
2600	66.33	68.55
2100	65.24	67.68
1600	64.39	65.81
1100	61.80	65.28
600	58.47	63.10

Table 6. 3

Acceptor and donor splice-site prediction 11ptAvg results. Recursive feature elimination is performed for mixed-features models of acceptor and donor sites. Each iteration we reduced the number of features by 200. For each case, we separated the structure- from sequence-based features and built separate prediction models for each. These results are also listed.

Acceptor Models (No.Features and 11ptAvg)

Mix Model		Structure		Sequence	
5848	89.74	2941	66.55	2907	90.35
5000	90.05	2400	64.23	2600	90.02
4400	90.76	1981	62.83	2419	90.27
4000	90.55	1668	60.26	2332	90.26
3400	90.37	1227	58.52	2173	90.09
3000	90.36	957	55.41	2043	89.69
2400	90.25	583	45.84	1818	89.68
2000	89.51	376	37.60	1625	89.30
1400	89.12	153	32.04	1248	88.51
1000	88.42	57	24.00	943	87.79

Donor Models (No. Features and 11ptAvg)

Mix Model		Structure		Sequence	
4823	89.46	3148	-	1675	90.61
4000	89.83	2482	64.68	1518	90.22
3400	90.13	2009	62.11	1391	90.26
3000	90.36	1679	60.42	1321	90.19
2400	90.76	1206	57.00	1194	90.20
2000	90.75	933	50.58	1067	90.23
1600	90.57	677	44.25	923	90.13
1000	90.15	335	34.08	665	89.82
600	89.46	183	25.64	417	89.20

Table 6. 4

The FGA-generated position-specific paired 5-mer features that overlapped with FasESS and ESR sets. The features are grouped into six regions: far, near, and close upstream or downstream from the splice-site location.

Region	5-mer Features overlapping with Fas-ESS signals
Far - upstream	CCTGG, GCTGC, TGCTG, TTGTG
Near - upstream	CCCTG, CCTGC, CCTGG, CCTTC, CGAGG, CGTGG, GCCAT, GCGGC, TGGAG
Close - upstream	CCAGG, CCAGT, CCATC, CCTGG, CTGCA, CTTCC, GGCAA
Close - downstream	AAGTT, AGATG, AGATT, AGGTG, AGGTG, AGTAT, AGTGA, AGTTG, AGTTT, GTTCT, GTTCT, AGGGG GGTAG, GGTGT, GTATA, GTTCA, GTTGT, GTTTG, GTTTT, AAGGG, AAGTG, GTTGG, TGGGA, CTGGG AGGGT, AGGTA, AGTAG, AGTCC, AGTGG, AGTTA, GATTA, GTAGG, GTGGC, TGGGG, TTTCT, GGGGG
Near - downstream	GAGGG, GGGAG, GGGGA, GGGTG, GTGGG, CGGGG, GGAGG, GGGGT, GGTGG, GGGGG, CTGGG, AGGGG

6.8 Figures of Chapter 6

Figure 6.1. Secondary-structure sequence examples for acceptor splice site (S_1 and S_2), as outputted by Afold.

The acceptor-site consensus "AG" is at positions [80,81] in the sequence. The sequences consist of 162 letters each from the nucleotide alphabet {A, C, G, T}. The upstream region of the sequence is composed of the 80 nucleotides, shown in blue, and the downstream region consists of 80 nucleotides, shown in green. The nucleotides which the Afold algorithms has predicted to be paired in the secondary structure are shown in upper case, and the unpaired nucleotides are shown in lowercase.

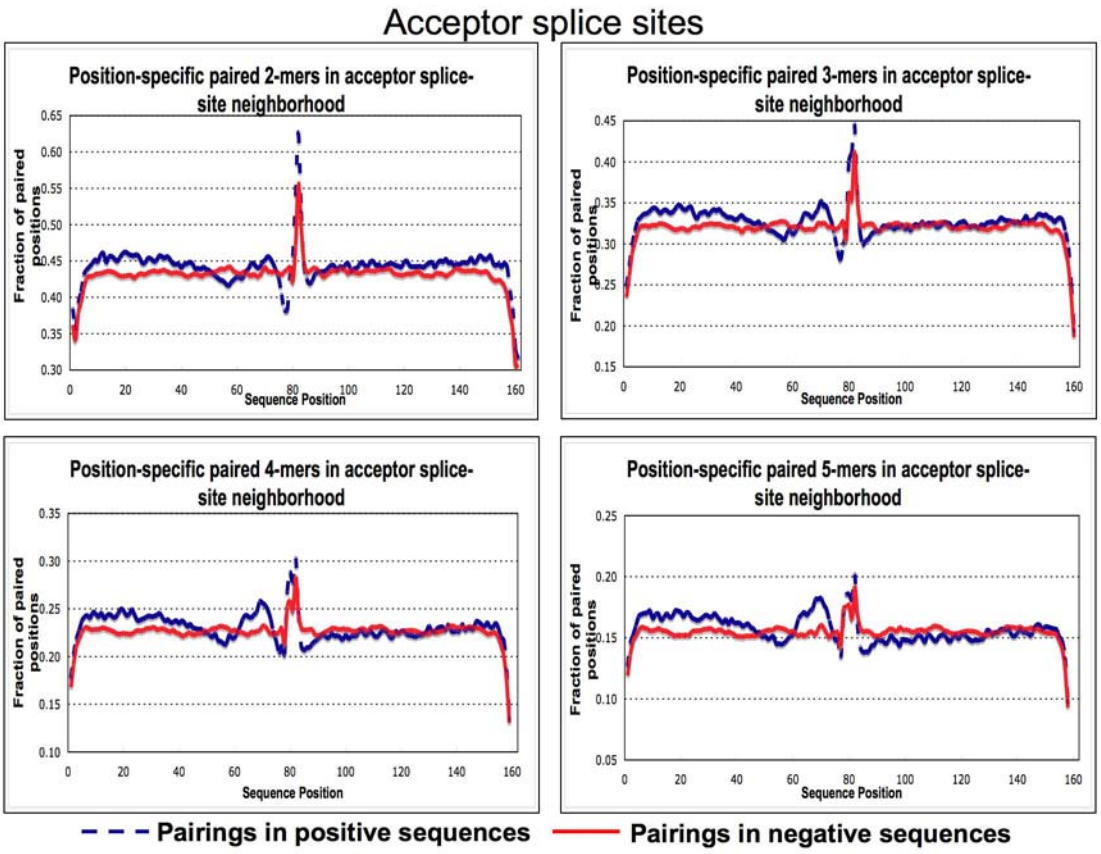


Figure 6.2**.** Position-specific paired features found in true acceptor-site sequences (positive) vs. non-acceptor-site sequences (negative).

The acceptor-site consensus "AG" is at positions [80,81] in the sequence. The upstream region, the sequence region to the left of the splice site, indicated pairing affinity in the true sequences.

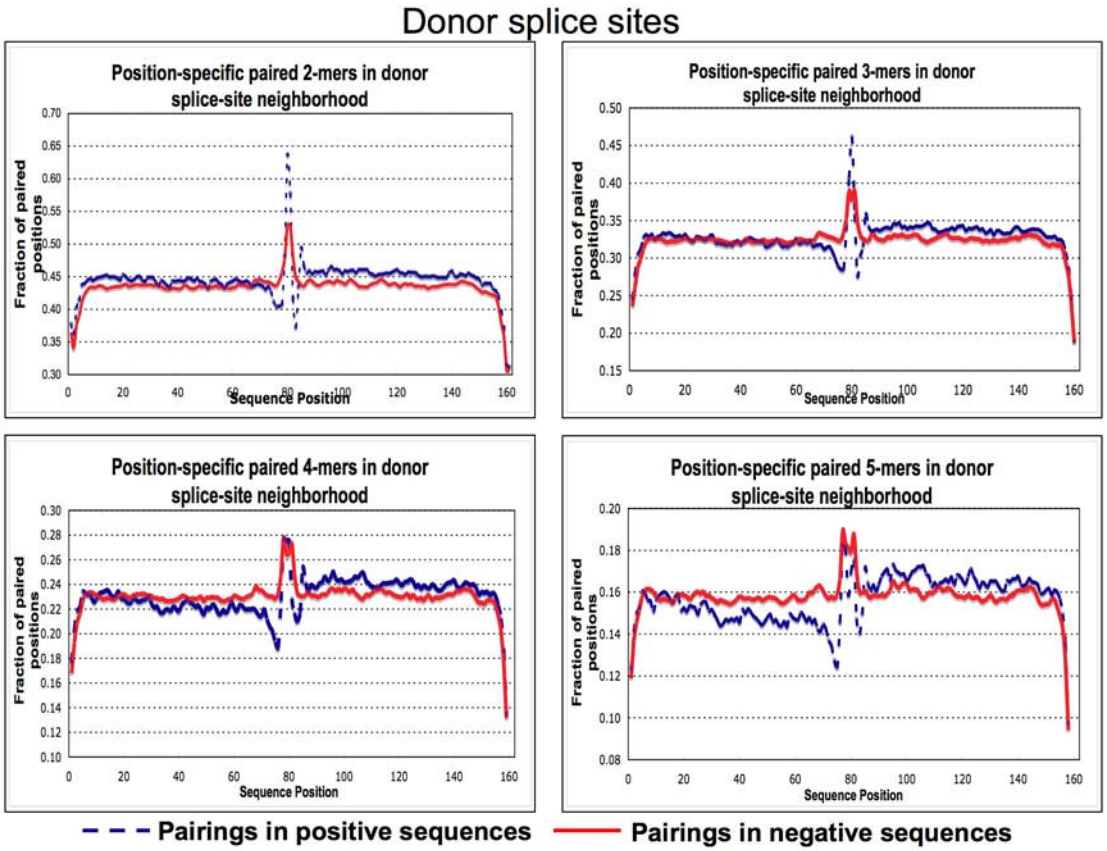


Figure 6.3**.** Position-specific paired features found in true donor-site sequences vs. non-donor-site sequences.

The donor-site consensus "GT" is at positions [80,81] in the sequence. The upstream region shows a lower pairing affinity, compared to the downstream region, the sequence region to the right of the splice site. A smaller fraction of pairings was observed in true sequences, compared to negative sequences in the upstream region.

Bibliography

1. Arita M, Tsuda K, Asai K. Modeling splicing sites with pairwise correlations. Bioinformatics, 18 Suppl 2:27-34, 2002.
2. Baten AKMA, Chang BCH, Halgamuge SK, Li J. Splice site identification using probabilistic parameters and svm classification. BMC Bioinformatics, 7(S5), 2006.
3. Berget SM, Moore C, Sharp PA. Spliced segments at the 5' terminus of adenovirus 2 late mRNA. Proc Natl Acad Sci U S A, 74(8):3171-3175, Aug 1977
4. Black DL, Chabot B, Steitz JA. U2 as well as U1 small nuclear ribonucleoproteins are involved in premessenger RNA splicing. Cell, 42(3):737-750, Oct 1985.
5. Blum AL, Langley P: Selection of relevant features and examples in machine learning. Artificial Intelligence 1997, 245-271.
6. Brendel, V. and Kleffe, J. (1998) Prediction of locally optimal splice sites in plant pre-mRNA with applications to gene identification in Arabidopsis thaliana genomic DNA. Nucleic Acids Res. 26, 4748-57.
7. Brunak S, Engelbrecht J, Knudsen S. Prediction of human mRNA donor and acceptor sites from the DNA sequence. J Mol Biol, 220(1):49-65, Jul 1991.
8. Burge C, Karlin S. Prediction of complete gene structures in human genomic dna. Journal of Computational Biology, 1997
9. Cartegni L, Hastings ML, Calarco JA, de Stanchina E and Krainer AR. (2006) Determinants of exon 7 splicing in the muscular atrophy genes SMN1 and SMN2. Am J Hum Genet 78:63-77.
10. Cartegni L, Wang J, Zhu Z, Zhang MQ, Krainer AR: ESEfinder: A web resource to identify exonic splicing enhancers. Nucleic Acids Res. 2003, 1;31(13):3568-71.
11. Chow LT, Gelinas RE, Broker TR, Roberts RJ. An amazing sequence arrangement at the 5' ends of adenovirus 2 messenger RNA. Cell, 12(1):1-8, Sep 1977.

12. Chua K, Reed R: An upstream AG determines whether a downstream AG is selected during catalytic step II of splicing. Mol. Cell. Biol., 2001. 5:1509-14.
13. Churbanov AE, Rogozin IB, Deogun JS, Ali HH. Method of predicting Splice Sites based on signal interactions. Biol Direct, 1(1), Apr 2006.
14. Coulter LR, Landree MA, Cooper TA: Identification of a new class of exonic splicing enhancers by in vivo selection. Mol. Cell. Biol. 1997. 17(4):2143-2150.
15. Crooks GE, Hon G, Chandonia JM, Brenner SE: WebLogo: A sequence logo generator. Genome Res. 2004, 14:1188-1190.
16. Degroeve S, De Baets B, Van de Peer Y, Rouze P: Feature subset selection for splice site prediction. Bioinformatics 2002, 18 Suppl 2:S75-83.
17. Degroeve S, Saeys Y, De Baets B, Rouze P, Van de Peer Y: SpliceMachine: predicting splice sites from high-dimensional local context representations. Bioinformatics 2005, 21(8):13
18. Fairbrother WG, Yeo GW, Yeh R, Goldstein P, Mawson M, Sharp PA, Burge CB: RESCUE-ESE identifies candidate exonic splicing enhancers in vertebrate exons. Nucleic Acids Res. 2004, 1;32(Web Server issue):W187-90.
19. Fickett JW. The gene identification problem: an overview for developers. Computers Chem, 20:103-118, 1996.
20. Gooding C, Clark F, Wollerton MC, Grellscheid SN, Groom H, Smith CWJ: A class of human exons with predicted distant branch points revealed by analysis of AG dinucleotide exclusion zones. Genome Biol. 2006, 7:R1.
21. Goren A, Ram O, Amit M, Keren H, Lev-Maor G, Vig I, Pupko T, Ast G: Comparative analysis identifies exonic splicing regulatory sequences-the complex definition of enhancers and silencers. Mol. Cell 2006, 23;22(6):769-81.
22. Guigo R, Flicek P, Abril JF, Reymond A, Lagarde J, Denoeud F, Antonarakis S, Ashburner M, Bajic VB, Birney E, Castelo R, Eyras E, Ucla C, Gingeras TR, Harrow J, Hubbard T, Lewis SE, Reese MG: EGASP: the human ENCODE Genome Annotation Assessment Project. Genome Biol. 2006, 7 Suppl 1:S2.1-31.

23. Han K, Yeo G, An P, Burge CB, Grabowski PJ: A combinatorial code for splicing silencing: UAGG and GGGG motifs. PLoS Biol. 2005, 3:e158.
24. Hastings ML, Allemand E, Duelli DM, Myers MP, Krainer AR: Control of pre-mRNA splicing by the general splicing factors PUF60 and U2AF65. PLoS ONE. 2007, 2:e538.
25. Hebsgaard, S.M., Korning, P.G., Tolstrup, N., Engelbrecht, J., Rouze, P. and Brunak, S. (1996) Splice site prediction in Arabidopsis thaliana DNA by combining local and global sequence information, Nucleic Acids Res., 24, 3439-3452.
26. Islamaj Dogan R, Getoor L, Wilbur WJ, A feature generation algorithm with applications to biological sequence classification, Chapter in Computational Methods of Feature Selection, Huan Liu and Hiroshi Motoda editors. (2007).
27. Islamaj Dogan R, Getoor L, Wilbur WJ, Mount SM, Features generated for computational splice-site prediction correspond to functional elements, BMC Bioinformatics, (2007).
28. Islamaj Dogan R, Getoor L, Wilbur WJ, Mount SM, SplicePort: an interactive splice-site analysis tool, Nucleic Acids Research, (June 2007).
29. Islamaj R, Getoor L, Wilbur WJ: A feature generation algorithm for sequences with application to splice-site prediction. In Proceedings of European Conference on Principles and Practice of Knowledge Discovery in Databases 2006:553-560.
30. Kohavi R, John G: The wrapper approach. In Feature Extraction, Construction and Selection : A Data Mining Perspective. Edited by Liu and Motoda; 1998, 33-48.
31. Kol G, Lev-Maor G, Ast G: Human–mouse comparative analysis reveals that branch-site plasticity contributes to splicing regulation. Hum. Mol. Genet. 2005. 14(11):1559-1568
32. Koller D, Sahami M: Toward optimal feature selection. In Proc. 13th Intern. Conf. on Machine Learning 1996, 284-292.
33. Královicová J, Vorechovsky I: Position-dependent repression and promotion of DQB1 intron 3 splicing by GGGG motifs. J Immunol. 2006, 176 (4):2381-8
34. Krawczak, M., Reiss, J., Cooper, D.N., The mutational spectrum of single base-pair substitutions in mRNA splice junctions of human genes: causes and consequences. Hum Genet, 90(1-2):41 (1992).

35. Ladd AN, Cooper TA: Finding signals that regulate alternative splicing in the post-genomic era. Genome Biol. 2002, 3(11).
36. Lim LP, Burge CB: A computational analysis of sequence features involved in recognition of short introns. Proc. Natl. Acad. Sci. 2001, 98(20):11193-8.
37. Liu H, Wong L: Data Mining Tools for Biological Sequences. Journal of Bioinformatics & Computational Biology 2003, 1(1):139-168.
38. Mathe C, Sagot MF, Schiex T, Rouze P: Current methods of gene prediction, their strengths and weaknesses. Nucleic Acids Res. 2002, 30(19):4103-4117.
39. McCullough AJ, Berget SM: An intronic splicing enhancer binds U1 snRNPs to enhance splicing and select 5' splice sites. Mol. Cell Biol. 2000, 20:9225–9235.
40. McCullough AJ, Berget SM: G triplets located throughout a class of small vertebrate introns enforce intron borders and regulate splice site selection. Mol. Cell Biol. 1997, 17:4562–4
41. Mount SM, Pettersson I, Hinterberger M, Karmas A, Steitz JA., The U1 small nuclear RNA-protein complex selectively binds a 5' splice site in vitro. Cell, 33(2):509{518, Jun 1983.
42. Nelson KK, Green MR: Mammalian U2 snRNP has a sequence-specific RNA-binding activity. Genes Dev. 1989, 3:1562-1571.
43. Ogurtsov AY, Shabalina SA, Kondrashov AS, Roytberg MA, Analysis of internal loops within the rna secondary structure in almost quadratic time, Bioinformatics, vol. 22, no. 11, 2006.
44. Patterson DJ, Yasuhara K, Ruzzo WL, Pre-mrna secondary structure prediction aids splice site prediction, in Pacific Symposium on Biocomputing, 2002.
45. Pertea M, Lin X, Salzberg S: GeneSplicer: a new computational method for splice site prediction. Nucleic Acids Res. 2001, 29(5):1185-90.
46. Pertea M, Mount SM, Salzberg SL: A computational survey of candidate exonic splicing enhancer motifs in the model plant Arabidopsis thaliana. BMC Bioinformatics 2007, 8:15
47. Salzberg SL. A method for identifying splice sites and translational start sites in eukaryotic mRNA. Comput Appl Biosci, 13(4):365-376, Aug 1997.

48. Schneider KM. A new feature selection score for multinomial naïve bayes text classification based on kl-divergence. In Meeting of the Association of Computational Linguistics (ACL), pages 186–189, 2004.
49. Singh R, Valcarcel J, Green MR: Distinct binding specificities and functions of higher eukaryotic polypyrimidine-tract binding proteins. Science 1995. 268:1173-1176.
50. Solovyev VV, Salamov AA, Lawrence CB. Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. Nucleic Acids Res, 22(24):5156-5163, Dec 1994.
51. Staden R. Computer methods to locate signals in nucleic acid sequences. Nucleic Acids Research, 12(1):505–519, 1984.
52. Stadler MB, Shomron N, Yeo GW, Schneider A, Xiao X, Burge CB: Inference of Splicing Regulatory Activities by Sequence Neighborhood Analysis. PLoS Genet. 2006, 2(11): e191
53. Staley JP, Guthrie C. Mechanical devices of the spliceosome: motors, clocks, springs, and things. Cell, 92(3):315-326, Feb 1998.
54. Sun H, Chasin LA. Multiple splicing defects in an intronic false exon. Mol Cell Biol, 20(17):6414-6425, Sep 2000.
55. Vignal A, Lisacek F, Quinqueton J, d'Aubenton Carafa Y, Thermes C. A multi-agent system simulating human splice site recognition. Comput Chem, 23(3-4):219-231, Jun 1999.
56. Wang Z, Rolish ME, Yeo G, Tung V, Mawson M, Burge CB: Systematic identification and analysis of exonic splicing silencers. Cell 2004, 119:831-845.
57. Witten IH, Moffat A, Bell TC. Managing gigabytes (2nd ed.): compressing and indexing documents and images. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.
58. Yang Y, Pedersen JP: A Comparative Study on Feature Selection in Text Categorization. In Proceedings of the Fourteenth International Conference on Machine Learning 1997, 412-420.
59. Yeo G, Burge C: Maximum entropy modeling of short sequence motifs with application to RNA splicing signals. J Comput. Biol. 2004, 11(2-3):377-94.

60. Yu L, Liu H, Feature selection for high-dimensional data: A fast correlation-based filter solution. In Machine Learning, Proceedings of the 20th International Conference (ICML 2003), pages 856–863, 2003.
61. Zhang L, Luo L. Splice site prediction with quadratic discriminant analysis using diversity measure. Nucleic Acids Research, 31(21):6214–6220, 2003.
62. Zhang MQ, Marr TG. A weight array method for splicing signal analysis. Computational Applications in Biological Sciences, 9(5):499–509, 1993.
63. Zhang MQ. Statistical features of human exons and their flanking regions. Human Molecular Genetics, 7(5):919–932, 1998.
64. Zhang T, Oles F: Text categorization based on regularized linear classification methods. Information Retrieval 2001, 4:5-31.
65. Zhang XH, Chasin LA: Computational definition of sequence motifs governing constitutive exon splicing. Genes Dev. 2004. 18:1241-50
66. Zhang XH, Heller KA, Hefter I, Leslie CS, Chasin LA: Sequence information for the splicing of human pre-mRNA identified by support vector machine classification. Genome Res. 2003, 13(12):2637-50.
67. Zhang XH, Kangsamaksin T, Chao MS, Banerjee JK, Chasin LA: Exon inclusion is dependent on predictable exonic splicing enhancers. Mol. Cell. Biol. 2005. 25: 7323-7332.
68. Zheng ZM: Regulation of alternative RNA splicing by exon definition and exon sequences in viral and mammalian gene expression. J Biomed Sci. 2004, 11(4):538.