REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

SPLICE SITE PREDICTION USING MACHINE LEARNING

ELHAM PASHAEI

PhD. THESIS
DEPARTMENT OF COMPUTER ENGINEERING
PROGRAM OF COMPUTER ENGINEERING

ADVISER
PROF. DR. NİZAMETTİN AYDIN

İSTANBUL, 2017

REPUBLIC OF TURKEY

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

SPLICE SITE PREDICTION USING MACHINE LEARNING

A thesis submitted by Elham PASHAEI in partial fulfillment of the requirements for the degree of **DOCTORATE OF SCIENCE** is approved by the committee on 08.06.2017 in Department of Computer Engineering, Computer Engineering Program.

Thesis Adviser

Prof. Dr. Nizamettin AYDIN

Yıldız Technical University

Co- Adviser

Assist. Prof. Dr. Alper YILMAZ

Yıldız Technical University

Approved By the Examining Committee

Prof. Dr. Nizamettin AYDIN

Yıldız Technical University _____

Prof. Dr. Banu DİRİ, Member

Yıldız Technical University _____

Assist. Prof. Dr. Arzucan ÖZGÜR, Member

Bogaziçi University _____

Assoc. Prof. Dr. Songül ALBAYRAK, Member

Yıldız Technical University _____

Prof. Dr. Zümray Dokur ÖLMEZ, Member

İstanbul Technical University _____

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my advisor Prof. Nizamettin Aydin for continuous support of my Ph.D. study, for his patience and motivation. His guidance helped me in all the time of research and writing of this thesis.

Next, I would like to thank my beautiful mother, who means the world to me. She was always supporting me and encouraging me with her best wishes. Without her, I never have gotten to where I am today.

At the end, I need to thank my generous father, who is my guardian angel in this world.

June, 2017

Elham PASHAEI

TABLE OF CONTENTS

Page

LIST OF ABBREVIATIONS vi
LIST OF FIGURES vii
LIST OF TABLES viii
ABSTRACT x
ÖZET xii

CHAPTER 1

INTRODUCTION 1

1.1 Literature Review 1
1.1.1 Splicing Mechanisms 2
1.1.2 Framework of Methods 2
1.1.3 Related Works 4
1.2 Motivation 11
1.3 Contribution of this Thesis 11
1.3.1 Novel DNA Encoding Methods using AdaBoost 12
1.3.2 RF for Feature Selection and Classification 13
1.3.3 A Novel DNA Encoding Method using SVM 14
1.3.4 Organization 14

CHAPTER 2

MATERIALS AND METHODS 16

2.1 Evaluation Datasets 16
2.2 Feature Extraction 17
2.2.1 Sparse Encoding Method 17
2.2.2 Position Weighted Matrix Model 17
2.2.3 Markovian Encoding Methods 18
2.2.4 Frequency Difference based Encoding Methods 20
2.2.5 Position Independent-Component based Encoding Method 21
2.2.6 Distribution of Triple Nucleotide Encoding method 22
2.3 Feature Selection 23
2.3.1 Fisher Score Feature Ranking Method 24
2.3.2 Random Forest Feature Ranking Method 24
2.4 Classification Methods 26
2.4.1 AdaBoost Classifier 26
2.4.2 SVM Classifier 27
2.4.3 NN Classifier 28

2.4.4 RF Classifier 29
2.5 Statistical Comparison 31
2.6 Implementation 31
2.7 Evaluation Criteria 31
2.8 Cross-Validation Design 32
2.9 Online Predictor Server 33

CHAPTER 3

PROPOSED APPROACHES 34

3.1 Novel Encoding Methods using AdaBoost 34
3.1.1 The DTMM1 Encoding Method 34
3.1.2 The FDDT Encoding Method 35
3.1.3 The SCMM1 Encoding Method 37
3.2 Performance of RF on Markovian Encoding Models 37
3.2.1 RF as Feature Ranking 38
3.2.2 RF as Classifier 38
3.3 A novel Encoding Method using SVM 39

CHAPTER 4

RESULTS 41

4.1 Experimental Results using AdaBoost 41
4.1.1 Evaluation on the HS3D Dataset 41
4.1.2 Optimum Value of Parameters 41
4.1.3 Performance Comparison on different Classifiers 44
4.1.4 Performance Comparison with different state-of-the-arts Methods 46
4.1.5 Evaluation on the NN269 Dataset 51
4.1.6 Online Prediction Server-HSSAda 53
4.2 Experimental Result using RF 54
4.2.1 Efficiency of RF as Feature Ranking Approach 54
4.2.2 Efficiency of RF as Classifier 55
4.3 Experimental Result using SVM 59

CHAPTER 5

DISCUSSION AND CONCLUSION 62

REFERENCES 65

CURRICULUM VITAE 72

LIST OF ABBREVIATIONS

AdaBoost	Adaptive Boosting
ANN	Artificial Neural Network
DA	Discriminative Analysis
DM	Distribution of triplet Nucleotides with Markovian Model
DMM2	Double Second order Markov Model
EI	Exon-Intron junction (donor splice site)
FD	Frequency Difference
FSS	False Splice Site
HM	Hierarchical Multi-classifier
HMM	Hidden Markove Model
IE	Intron-Exon junction (acceptor splice site)
MCM	Markov Chain Model
ML	Machine Learning
MLP	Multi-Layer Perceptron
MM0	Zero order Markov Model
MM1	First order Markov Model
MM2	Second order Markov Model
MN-FDTF	Mono Nucleotide-Frequency Difference between True and False
PN-FDTF	Pairwise Nucleotide-Frequency Difference between True and False
RF	Random Forest
SC	Sequence Compositional Features
SVM	Support Vector Machine
TSS	True Splice Site
WAM	Weighted Array Model
WMM	Weighted Matrix Model

LIST OF FIGURES

Page

Figure 1.1 Central dogma of biology 1
Figure 1.2 A pictorial description of splice sites 2
Figure 1.3 Splicing process in cells 3
Figure 1.4 Alternative splicing 3
Figure 1.5 Overall steps of the methods for predicting DNA splice sites 4
Figure 1.6 Flow diagram of eukaryotic splice type identification 5
Figure 1.7 The overall structure of the contribution 15
Figure 2.1 Example of the PWM model on several DNA aligned sequences [77] 18
Figure 2.2 An example of first order Markov model 19
Figure 2.3 Model description of MCM encoding method [61] 20
Figure 2.4 Overview of the process of the FDTD encoding methods [78] 21
Figure 2.5 Illustration of AdaBoost classifier 26
Figure 2.6 Maximum margin hyperplane and margins for an SVM trained with samples from two classes 28
Figure 2.7 Flowchart of SVM tuning process 29
Figure 2.8 Illustration of a random forest [93] 30
Figure 3.1 The F-score values on the features which were obtain by MN-encoding method on NN269 Donor sites 36
Figure 4.1 Evaluation of AdaBoost performance at different numbers of iteration on HS3D and NN269 datasets 42
Figure 4.2 Snapshot of the developed web tool 53
Figure 4.3 Global accuracy of different percentage of selected features using F-score feature ranking and random forest feature ranking methods on a) Balanced acceptor splice sites, b) Balanced donor splice sites, c) Unbalanced acceptor splice sites and d) Unbalance donor splice sites datasets for assessing performance of MM1-SVM method 55
Figure 4.4 Classification performance of the different state-of-the-art methods for both HS3D and NN269 datasets 58

LIST OF TABLES

Page

Table 1.1 Summary of most popular in silico tools for acceptor and donor splice site prediction with user-friendly web interface [42] 11
Table 4.1 Performance comparison of the SVM, RF, NN, and AdaBoost under three proposed encoding methods on predicting balanced Acceptor splice sites of HS3D datasets 43
Table 4.2 Performance comparison of the SVM, RF, NN, and AdaBoost under three proposed encoding methods on predicting unbalanced Acceptor splice sites of HS3D datasets 44
Table 4.3 Performance comparison of the SVM, RF, NN, and AdaBoost under three proposed encoding methods on predicting balanced Donor splice sites of HS3D datasets 45
Table 4.4 Performance comparison of the SVM, RF, NN, and AdaBoost under three proposed encoding methods on predicting unbalanced Donor splice sites of HS3D datasets 46
Table 4.5 Performance comparison of the proposed methods with others state of art methods on balanced Acceptor splice sites 47
Table 4.6 Performance comparison of the proposed methods with others state of art methods on unbalanced Acceptor splice sites 48
Table 4.7 Performance comparison of the proposed methods with others state of art methods on balanced Donor splice sites 49
Table 4.8 Performance comparison of the proposed methods with others state of art methods on unbalanced Donor splice sites 50
Table 4.9 Performance comparison of SVM, RF, NN, and AdaBoost under three proposed encoding methods on predicting Acceptor splice sites of NN269 dataset 51
Table 4.10 Performance comparison of SVM, RF, NN, and AdaBoost under three proposed encoding methods on predicting Donor splice sites of NN269 dataset 52
Table 4.11 Comparison of different models on NN269 dataset 52
Table 4.12 Performance comparison with other in-silico tools using the independent Acceptor test set 53
Table 4.13 Performance comparison with other in-silico tools using the independent Donor test set 54
Table 4.14 Comparison of classification performance of SVMs and RFs using Markovian encoding methods 56
Table 4.15 Performance comparison of the DMM2-SVM with others state of art methods on balanced Acceptor splice sites 60

Table 4.16 Performance comparison of the DMM2-SVM with others state-of-the-art methods on unbalanced Acceptor splice sites 60
Table 4.17 Performance comparison of the DMM2-SVM with others state-of-the-art methods on balanced Donor splice sites..... 60
Table 4.18 Performance comparison of the DMM2-SVM with others state-of-the-art methods on unbalanced Donor splice sites 61

ABSTRACT

SPLICE SITE PREDICTION USING MACHINE LEARNING

Elham PASHAEI

Department of Computer Engineering

Ph.D. Thesis

Adviser: Prof. Nizamettin AYDIN

Co-adviser: Assist. Prof. Alper YILMAZ

Due to an explosion in the quantity of DNA sequences over the past decades, development of new methods to accurately detect the genes is vital. The success of these methods strongly depends on precise identification of the splice sites.

In eukaryotic genomes, each gene is composed of exons and introns. During DNA transcription only exons of the gene, which contain codes for proteins are transcribed into mRNAs. The term splice site refers to the boundary between exon and intron. While the intron-exon junction with consensus dinucleotide AG is called acceptor splice site, donor splice site refers to an exon-intron junction with consensus dinucleotide GT. In DNA sequence, splice site prediction is a search problem for finding donor and acceptor boundaries.

Numerous Machine Learning methods have been used for splice sites identification. Performances of these methods highly depend on the DNA encoding approaches, which try to extract informative features from DNA sequences.

Using AdaBoost classifier, we have proposed three new DNA encoding methods for feature extraction by combining several approaches that have already proven successful in determining pattern around splice sites. the proposed approaches provided significantly better performance than eleven current state-of-the-art algorithms based on several performance criteria.

We also have developed an online prediction server (HSSAda) based on proposed approach, which is freely available at https://pashaei.shinyapps.io/hssada. The HSSAda tool achieved higher accuracy while compared with the existing tools like NNplice, WMM, MM1, and MEM, using the independent test set. It is believed the proposed

methods can be helpful in discovering location and structure of eukaryotic genes due to their high prediction accuracy and simplicity.

We also assessed the performance of RF as classification and feature selection method in splice site prediction domain. The investigation tried to answer the question whether RF outperforms SVM, which is the most outstanding classification approach in splice site detection, using Markovian encoding methods or not.

Finally, we proposed another DNA encoding method using SVM and second order Markov model for splice site detection.

Key words: Gene detection, splice sites prediction, machine learning, DNA encoding

ÖZET

MAKİNE ÖĞRENMESİ KULLANARAK UÇBİRLEŞTİRME YERİ TAHMİNİ

Elham PASHAEI

Bilgisayar Mühendisliği Anabilim Dalı

Doktora Tezi

Tez Danışmanı: Prof. Dr. Nizamettin AYDIN

Eş Danışman: Assist. Prof. Dr. Apler YILMAZ

Son on yılda DNA dizileri miktarında ki olağanüstü artış nedeniyle, genlerin doğru tespit edilmesi için yeni yöntemlerin geliştirilmesi hayati önem taşımaktadır. Bu yöntemlerin başarısı, uçbirleştirme bölgelerinin kesin olarak tanımlanmasına bağlıdır.

Ökaryotik genomlarda, her gen eksonlar ve intronlardan oluşur. DNA transkripsiyonunda, sadece protein kopyalarını içeren genin ekzonları mRNA'lara aktarılır. Uçbirleştirme terimi, ekson ve intron arasındaki sınırı belirtir. Konsensüs dinükleotit AG ile yapılan intron-ekson birleşimine, alıcı uçbirleştirme bölgesi denirken, verici uçbirleştirme yeri, konsensüs dinükleotit GT ile ekson-intron birleşme noktasını belirtir. DNA dizisinde uçbirleştirme öngörüsü, verici ve alıcı sınırlarını bulmak için bir arama problemidir.

Uçbirleştirme yeri tespiti için çok sayıda makine öğrenmesi yöntemi kullanılmıştır. Bu yöntemlerin performansları, DNA dizilerinden bilgilendirici özellikler çıkarmaya çalışan DNA kodlama yaklaşımlarına büyük ölçüde bağlıdır.

AdaBoost sınıflandırıcısını kullanarak, uçbirleştirme alanlarının etrafında desen belirlemede zaten başarılı olduklarını kanıtlamış birkaç yaklaşımı birleştirerek özellik çıkarımı için üç yeni DNA kodlama yöntemi önerdik. Önerilen yaklaşımlar, çeşitli performans kriterlerine dayanan mevcut en gelişmiş 11 algoritmadan çok daha iyi bir performans sağlamıştır.

Ayrıca, https://pashaei.shinyapps.io/hssada adresinde erişilebilen, önerilen yaklaşım temelli bir çevrimiçi tahmin sunucusu (HSSAda) geliştirdik. HSSAda aracı, bağımsız test setini kullanan NNplice, WMM, MM1 ve MEM gibi mevcut araçlar ile

karşılaştırıldığında daha yüksek doğruluk elde etmiştir. Önerilen yöntemlerin, ökaryotik genlerin yerini ve yapısını keşfetmelerinde, tahminlerinin doğruluğunun ve basitliğinin yüksek olması nedeniyle yararlı olabileceği düşünülmektedir.

Ayrıca, RF'nin uçbirleştirme yeri tahmin alanındaki sınıflandırma ve özellik seçimi yöntemi olarak performansını değerlendirdik. Bu araştırmada, Markov kodlama yöntemleri kullanan RF'nin uçbirleştirme tespitinde, en başarılı sınıflandırma yaklaşımı olan SVM'den üstün olup olmadığını sorusu yanıtlanmaya çalışılmıştır.

Son olarak, uçbirleştirme yeri tespiti için SVM ve ikinci dereceden Markov modelini kullanan başka bir DNA kodlama yöntemi önerdik.

Anahtar Kelimeler: Gen algılama, uçbirleştirme yeri tahmin, makine öğrenimi, DNA kodlama

CHAPTER 1

INTRODUCTION

1.1 Literature Review

Biological sequence data has been increasing rapidly during the past few decades, so there is a crucial need for effective methods to detect genes. The success of these methods strongly depends on precise identification of the splice sites. In eukaryotic genomes, each gene is composed of exons and introns. During DNA transcription only exons of the gene, which contain codes for proteins are transcribed into mRNAs (See Figure 1.1).

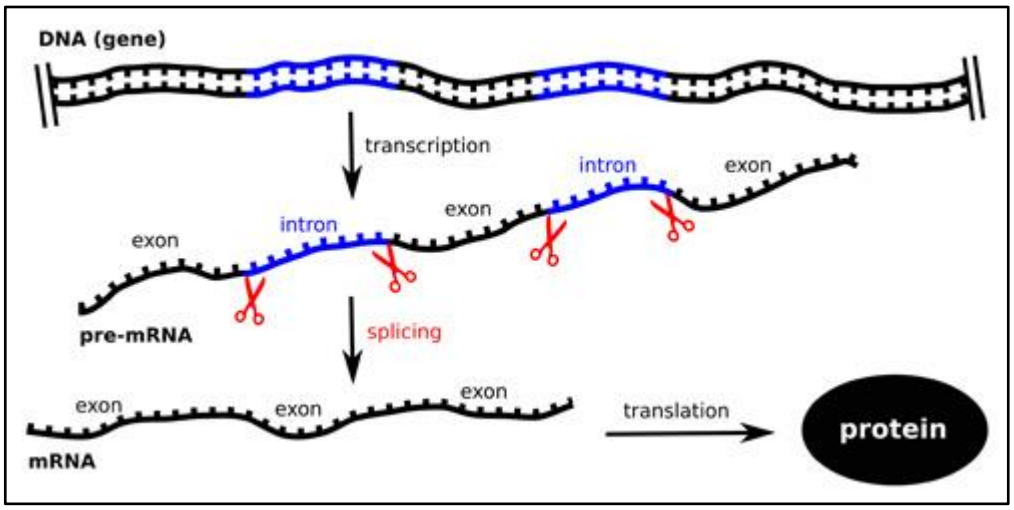

Figure 1.1 Central dogma of biology

The splice sites are known as the boundaries between exons and introns. The acceptor splice site is defined as transition site from intron to exon and distinguished by dinucleotide AG. The donor splice site is defined as transition site from exon to intron and distinguished by dinucleotide GT (See Figure 1.2). Large numbers of AG-GT consensus sites exist in the eukaryotic genes, but only 0.1%~1% of them are true splice sites [1]. In this respect, the splice site prediction is a search problem of identifying

whether AG/GT dinucleotide is true splice site or false site, which is known as one of the most important and challenging tasks in bioinformatics [1], [2], [3].

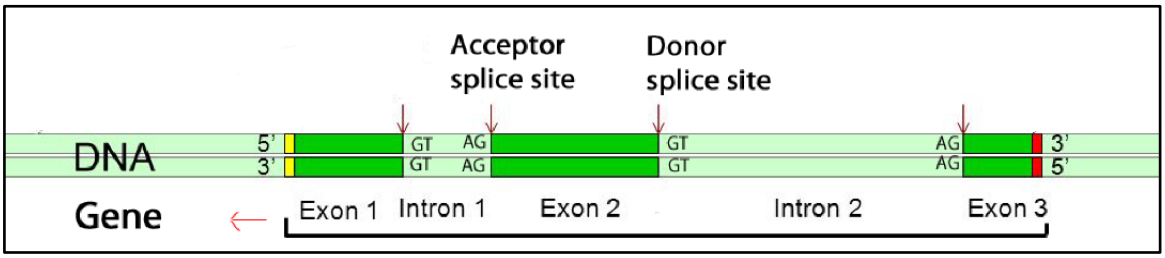

Figure 1.2 A pictorial description of splice sites

1.1.1 Splicing Mechanisms

Spliceosomes regulate and control splicing mechanism in the cell in order to make various proteins. They are composed of enzymes which called small nuclear ribonucleoproteins (snRNPs). The snRNPs identify conserved sequences in splice sites (AG and GT dinucleotide) and remove the introns. Then, they paste the exon together. Figure 1.3 shows the mechanism of splicing.

Sometimes splicing produces one protein for one gene by putting the exons together in one way. While alternative splicing allows the exons to be put together in different ways to generate multiple proteins from a single gene (See Figure 1.4). More than 5% of all genes can produce variant proteins by alternative splicing. A fine-tuned balance of factors regulate splice site selection. In [4], it has been discussed that how changes in alternative splicing can be a cause or consequence of human diseases.

1.1.2 Framework of Methods

To predict the splice site, approximately all of the proposed methods consist of three main steps; proper encoding scheme (feature extraction), feature selection (optionally), and classification. Machine learning methods are used to detect splice site (classification step). The input of machine learning classifiers is numerical, whereas the information of DNA sequences is given as strings. Therefore, encoding the DNA sequence into numbers is an initial and main task of splice site prediction (feature extraction step) [5]. The probabilistic encoding approaches such as the zero order Markov model (MM0), the first order Markov model (MM1), the second order Markov model (MM2), and the Markov Chain Model (MCM) are so famous and high usage methods [1], [6], [7], [8], [9], [10], [11], [12], [13], [14].

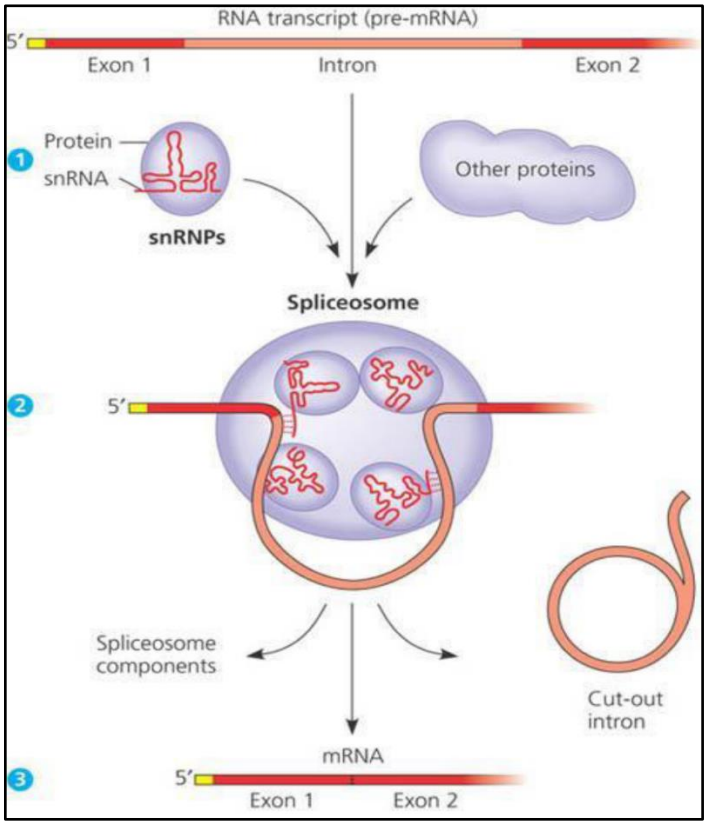

Figure 1.3 Splicing process in cells

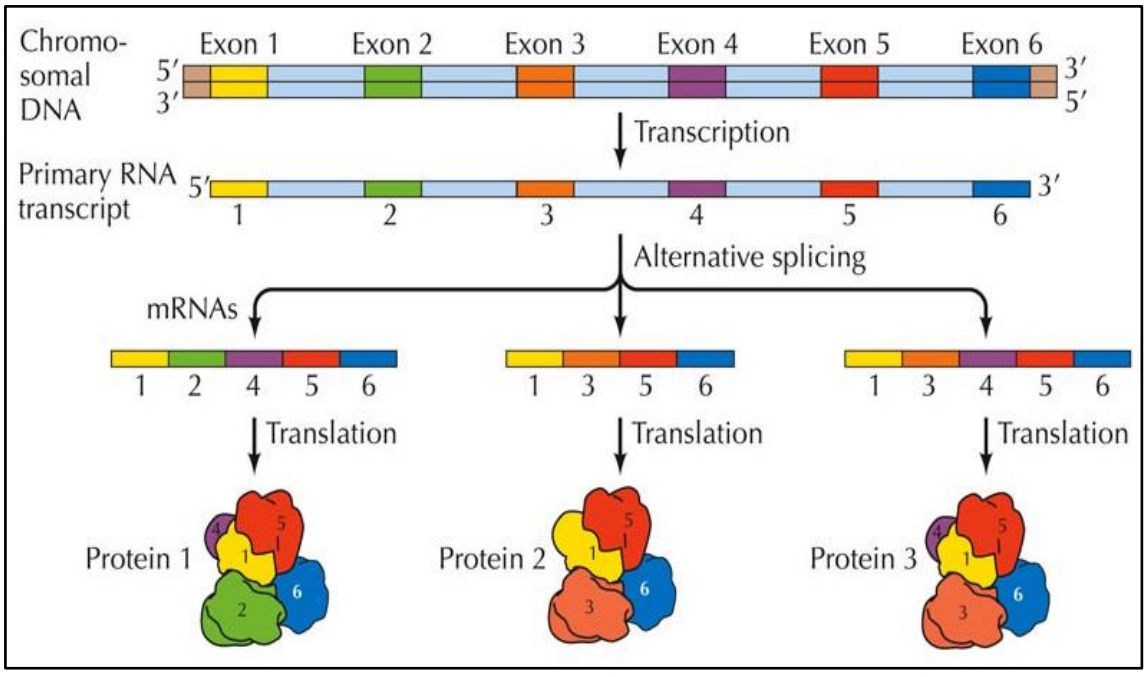

Figure 1.4 Alternative splicing

The problem of splice site detection is subdivided into two separate classification process- donor splice site (5′ss) prediction and acceptor splice site (3′ss) prediction. So,

two different models are constructed for each of them, which consist of three main steps. These processing steps are outlined in the following [13]:

- Feature extraction: Efficient DNA encoding schemes are used to provide as much information as the DNA sequences have.
- Feature selection: Choosing a relevant subset of features allows for a faster classification and better prediction accuracy.
- Classification: Machine Learning (ML) classifiers are trained on the provided features in order to discriminate true splice sites from false sites.

Steps of overall approaches for predicting splice sites have been described in Figure 1.5.

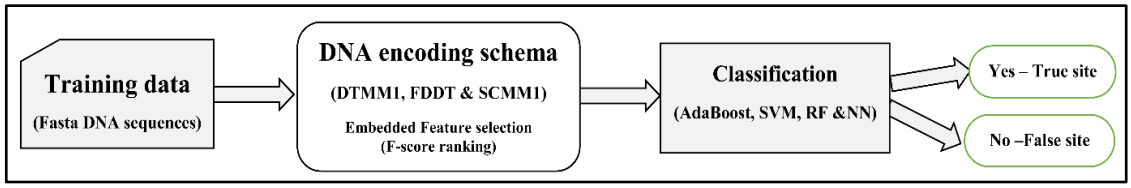

Figure 1.5 Overall steps of the methods for predicting DNA splice sites

1.1.3 Related Works

The problem of the splice site recognition in the literature has been categorized into two groups: splice type identification (STI) and splice site prediction (SSP). For any given DNA sequences, the STI considers a classification system that predicts whether the sequence belongs to an intron-exon (IE) boundary, exon-intron (EI) boundary or neither of them. However, the SSP provides a classification system which identifies whether a GT/AG dinucleotide is a true splice site or not. There are few studies that review and classify the splice site prediction methods. In [15], the SVM-based methods have been investigated, while the neural network-based methods have been reviewed in [16].

1.1.3.1 Splice Type Identification

Given a position in the middle of a DNA sequence, we try to decide whether this is an IE boundary, EI boundary or neither (N) (See Figure 1.6). To solve this problem, a knowledge-based artificial neural network (KBANN) proposed in [17]. It is a hybrid method which feeds inference propositional rules from the biological domain into the ANN. In [18], the European Community StatLog project utilized various classification approaches such as k-NN, C4.5, CART, NaïveBayes for primate splice site identification.

A batch-relevance based artificial intelligent (BRAIN) algorithm was introduced in [19]. This approach after replacing each nucleotide in the DNA sequence with 4-bit binary (orthonormal sparse) encoding, computes a relevance coefficient for each attribute in an iteration way using Boolean classification rules. Then, the performance of the BRAIN was increased by combing it with the NN and discriminative analysis (DA) method.

In [20], an algorithm based on novel motif model and pattern matching was developed for classification and detection of donor splice junction. This motif model considers 10 motifs for each sequence. The length of each motif is not less than 6 nucleotide and contains GT. Then, a motif Library was constructed. According to the score of pattern matching approach, the classification was performed. In [21], a knowledge-based cascade correlation algorithm (KBCC) was described. This method was an extension of CC learning algorithm [22].

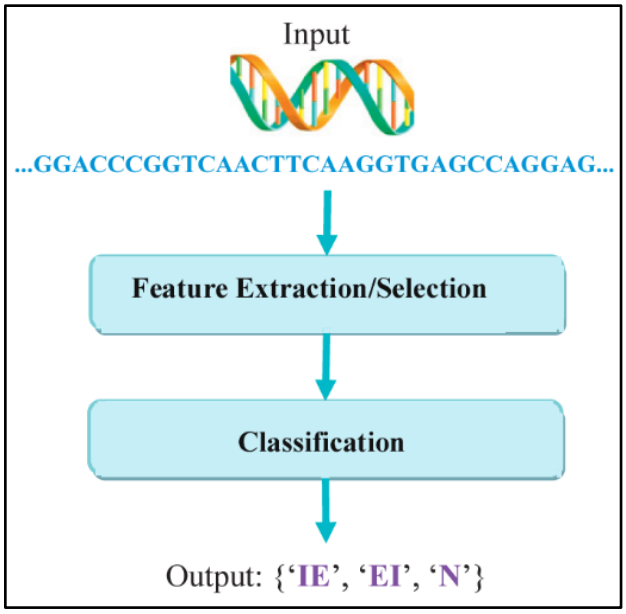

Figure 1.6 Flow diagram of eukaryotic splice type identification

The KBANN was able to find solve the problem by incorporating all the rules of determining splice junction, while KBCC showed that not all the rules are necessary in finding the best solution for splice-junction determination problem. A multiclass SVM method [23] was proposed for the Human splice type identification without any comparison with other works. Besides, the performance of the ensemble multiclass SVM was investigated. The results indicated that bagging SVM does not improve the accuracy. Since the input of SVM classifier is numerical, each nucleotide in the DNA

sequence was represented by a vector of binary values (A= (0 0 0 1), C = (0 0 1 0), G = (0 1 0 0) and T = (1 0 0 0)) in this study. In [24], effect of five noise-filtering technique which two of them can also identify redundant instances was investigated. Then, the obtained features were fed to multiclass SVM and decision tree C4.5. The best result was reported on multiclass SVM without preprocessing steps.

A hierarchical multi-classifier (HM) method [25] was defined by designing a four-stage pattern recognition scheme. After encoding each nucleotide using 8-bit binary, a feature ranking method and a dimensional reduction method were used to provide the input features. These features fed to SubSpace classifier, Edit Distance classifier, and Linear SVM in a hierarchical way, somehow the instances which rejected at each step are passed to the next classifier. The results of this method are superior to all methods in the literature. In [26], the SVM, Back-propagation NN and unsupervised Kohonen's Self-Organizing Map, (KSOM) approaches were employed for the recognition of splice junction sites in Human DNA sequences. It was demonstrated that the SVM yielded better prediction than others.

A new DNA encoding method based on k-mer frequency was proposed in [27]. In this study, the problem of splice site junction was considered in binary format which tried to distinguish IE from EI. The result showed that the 6-mer frequency of ACGT nucleotides when feeding to SVM light, which was extended with power series kernel, produced good accuracy. In [28], an unsupervised splice site prediction technique based on association analysis, namely assoDNA, was introduced. The method divides the training dataset into three subsets according to the class value and then generates frequent patterns. The advantage of the proposed method is that it can contain nucleotides at the arbitrary position in addition to the contiguous base sequences. The proposed method was compared with C4.5, naïve Bayes (NB), Instance-based method (with 10 nearest neighbors) and the SVM. From the results, the assoDNA produced a significantly high recall prediction.

In [29], a novel encoding method was proposed to deal with ambiguous values in the DNA sequences. The proposed encoding system used the 4-bit binary encoding for each nucleotide (A, C, T, G), while it used probability to encode ambiguous values (D, S, R, N) in DNA sequences. Then, a hyper decision tree structure (DTS) was used for classification in which K-Nearest Neighbors (KNN) and SVM were utilized in each

node. The KNN in this study used probability Hamming distance measure at the first node of the tree to distinguish the ‘N’ from EI and IE.

In [30], a new method was introduced based on genetic algorithm (GA) and a new version of Naive Bayes (NB) classifier, namely averaged one-dependence estimators with subsumption resolution (AODEsr). It works without any prior knowledge by utilizing the GA for selecting informative features and the AODEsr for the classification task. In [31], an NN based tree-structured pattern classifier, namely NNTree, was introduced. The method utilized multilayer perceptron (MLP) with back-propagation (BP) algorithm at each node of the tree. A new criterion was used as a splitting criterion instead of using the information gain ratio. The performance of the NNTree compared with different classification algorithms, namely Bayesian, C4.5, MLP, and cellular automata (CA). The results showed that the classification accuracy of the NNTree was higher than the others.

The UCI Molecular Biology (Splice-Junction Gene Sequences) Data Set [32] is a standard dataset which contains 3190 patterns. The length of each pattern is 60 nucleotides. Approximately, all of the mentioned works have utilized it for evaluating and comparing their approaches. According to the results that have been reported in the literature, the performance ranking of the methods from better to worse is: HM> DTS> multiclass SVM> AODEsr> NNTree> NNBRAIN > BRAIN> KBANN> MLP> ID3> NN. This ranking has been provided by considering test error rates.

1.1.3.2 Splice Site Prediction Methods

To solve the splice site prediction problem, we try to distinguish real splice sites (AG or GT) from the bulk of pseudo splice sites (non-sites). Since the dinucleotide AG or GT that correspond to real splice sites in the DNA sequence are less than 1%, this problem is one of the most challenging tasks in bioinformatics.

A large number of computational methods are available in order to accurately detect the splice sites. The weight matrix method (WMM) [33], [34] and weight array method (WAM) [35], also known as MM1, are the earliest and most weighty methods which utilize positional features and a threshold to predict splice sites. Baten [7] improved prediction accuracy of MM1 by using it as a powerful DNA encoding method and feeding the produced features to SVM (MM1-SVM). Another similar method is Reduced MM1-SVM [1] which combines only informative features with the SVM. The

Fisher Score (F-score) feature ranking method [36] was used for feature selection in this approach. Zhang [5] used a Bayes feature mapping as DNA encoding method and merge it with the linear SVM (B-SVM) to increase accuracy and decrease time complexity of splice site prediction. The disadvantage of this encoding method is that dependency between nucleotides is not considered. Huang [37] proposed four different encoding approaches, which were mono nucleotide (MN), pairwise nucleotide (PN), the combination of the MN with the frequency difference between true and false sites (FDTF), and the combination of the PN with the FDTF. The SVM classifier was applied to predict splice sites. The experimental results showed that PN-FDTF had the best performance among the others. A modified version of PN-FDTF encoding method was combined with RF (P-RF), which exhibited good performance in prediction of donor sites [38]. Sonnenburg [3] utilized SVM classifier without encoding methods for splice sites detection by applying weighted degree kernel (WD) and shift weighted degree kernel (WDS). In [11], a length variable Markov model (LVMM) was proposed. The method utilizes second order Markov model (MM2) for extracting DNA features and can choose a subset of informative features by considering the ratio of likelihood at each position. The proposed method produces good accuracy but determining the threshold parameters of the approach is a difficult task. In [13], the MM2 encoding method was combined with the SVM and principle feature analysis was used as feature selection method (MM2F-SVM). In [39], the SVM was replaced by Random Forest classifier (MM2-RF) and the accuracy was highly increased. However, considering high Markov model in the DNA encoding makes the approach complex and unrealistic [40]. Another DNA encoding method which is known as MCM is a combination of the MM1 and the MM2 [10]. In this method, the NN was used as the classifier. In [14], the SVM was employed instead of the NN and accuracy of the approach was improved (MCM-SVM). The MCM encoding method divides the sequences into three segments. However, this division has a direct effect on the performance of the method and it is not easy work. Wei [41] proposed three novel DNA encoding approaches which were orthogonal encoding, codon encoding, and sequential information. In this work, the linear SVM was used as the classifier (ECS-LSVM).

In [12], the sequential information under the name of distribution of tri-nucleotides was merged with MM1 encoding method using both true and false sites (DM). F-score feature ranking method was used to select informative feature and the SVM was

employed for classification task (DM-SVM). In [40], new encoding methods based on the hybrid of density information of each nucleotide along with positional and chemical property (DPCH-SVM) were proposed. The MSC+Pos(+APR)-SVM [2] approach, which is hybrid of multi scale sequence component (MSC) encoding, position features (Pos) and adjacent position relationship (APR) encoding method along with the SVM, is the most outstanding method in Human splice site prediction domain with regard to its high-throughput accuracy. However, the number of useless features that are produced by the MSC encoding alone is too many. Consequently, the method requires high computational cost [2]. Also, the associated encoding approach requires several parameters tuning which impose additional time complexity to the method.

1.1.3.3 In-Silico Tools on Splice Site Detection

There are lots of in silico tools that predict potential splice sites (Table 1.1). Recently, the main application of these tools has changed from the precise recognition of splicing signals to detect the transcriptional impact of the mutation on known splice sites due to high dependency (up to 15%) of human genetic diseases to splicing defects [42].

Most tools receive single or multi-DNA sequences as the input with or without specifying the position of splice sites. The length of the sequences should be fixed by the user in the former condition whereas in the latter condition potential splice sites are seek automatically through the whole length of the input sequence. The output of the prediction tools should be precisely interpreted when are employed in clinical practice. Most tools produce a score for exhibiting the potential of the predicted splice sites. A higher score always indicates a higher probability of a site being a true splice site. A brief description of the algorithms which have been employed in the tools is useful for understanding their advantages and disadvantages.

In 1987, Shapiro and Senapathy [43] identified splice sites by scoring and ranking the DNA sequences using k-mer position weighted matrix model (PWM) which was used in Splice-Site Analyzer Tool. In contrast to PWM which assume independence among all positions, an improved PWM, which consider higher nucleotides frequency in certain site positions and the mutual correlation between nucleotides of different site positions, was introduced by Rogozin and Milanesi to produce SpliceView tool [44].

Backpropagation NN was first addressed by Brunak et al [45] for recognition of splice sites. They encoded the sequence of nucleotides using 4-bit sparse schema and gave

them as the input to the one hidden layer NN (NetGene). The method has utilized a threshold and a cutoff level on the outputs in order to classify splice sites. This cut-off level was obtained by prediction of the coding region based on the base compositions (codon usage) to control false positive rate in the proposed method. Due to the combination of the NN with predicted coding region, it is called a joint prediction schema. NetPlantGene tool, which is a combination of the aforementioned method with a rule-based system in order to increase accuracy, was used to predict splice sites of plants DNA sequences in ref [9]. This method consists of two steps. The first step is prediction step where a global detection of coding regions regulate a cut-off level for local prediction of splice sites utilized hybrid of sparse encoding and NN. The second step is refinement step using rules that obtained by analysis of mistakes which occur at the first step [16]. A further improvement in acceptor splice site in the previous method was obtained by incorporating branch points to reduce false positive rates in NetGene2 tool [46]. HMM was used to predict branch points and another NN was added to the model to increase the performance of the model.

NNSplice is another NN based splice site identification method which has been utilized in Genie program for gene finding [16]. A 4-bit sparse encoding method with a window size of 10 and 40 nucleotides for donor and acceptor site respectively, was fed to back propagation NN to predict splice sites. Both true and false splice sites were used to train the NN. Later on, Reese et al [8] improved the model by the hybrid of the NN with 16-bit code per nucleotide pair encoding scheme.

The maximal dependence decomposition model, which was incorporated in GENESCAN [6], is a decision tree based model that captures strong dependencies between adjacent and nonadjacent position. Genesplicer [49] improved the previous model using Markov model. Spliceport [51] is another tool that utilizes SVM classifier to predict splice site. It used feature generation algorithm that automatically captured important sequence-based features for feeding to SVM. Finally, MaxEntScan [50] is another tool that used maximum entropy model (MEM) to predict splice sites.

Table 1.1 Summary of most popular in silico tools for acceptor and donor splice site prediction with user-friendly web interface [42]

Tool	Input	Output
SS Analyzer Tool [43]	Single/multiple sequences (5′: 9bp (−3 to +6); 3′: 15bp (−14 to +1))	S & S score (0–100)
HSPL-RNASPL [47]		
NetPlantGene [9]	-	-
SplicePredictor [48]	Single/multiple sequences	*-Value (3–15) determined by P, ρ, and γ values
SpliceView [44]	Single sequence (Length ≤31,000 bp)	S &S score (0–100)
NNSplice0.9 [8]	Single/multiple sequences	Score (0–1)
GeneScan [6]	Single sequence (Length≤1 million bp)	Probability score (0–1)
NetGene2 [46]	Single sequence (200bp < length < 80,000bp)	Confidence score(0–1)
GeneSplicer [49]		
MaxEntScan[50]	Single/multiple sequences (5′: 9bp; 3′:23bp)	
SplicePort [51]	Single/multiple sequences (Length ≤30,000 bp)	Feature generation algorithm score
Human Splice Finder[52]	Single sequence (Length ≤5,000 bp)	S & S score (0–100)

1.2 Motivation

In the past few decades, numerous methods have been proposed to detect the splice sites, such as hidden Markov model (HMM) [11], [53], Bayesian networks [54], [55], artificial neural network (ANN) [56], [57], support vector machine (SVM) [1], [3], [7], [12] and decision trees [58], [59]. However, due to complex dependencies existing among the bases around splice sites, the splice site prediction is still a difficult problem, i.e., splice site prediction is still a major bottleneck in gene finding [60]. Thus, development of new methods to accurately predict the splice sites is important.

1.3 Contribution of this Thesis

Generally, the splice site prediction methods are composed of feature extraction and classification approaches. DNA encoding methods are used for feature extraction to provide the characteristics of the DNA sequences, while ML methods are employed for classification part [41]. An efficient DNA encoding method leads to better classification accuracy in predicting splice sites [40]. There are many computational approaches for identifying splice sites, such as SVM [12], [14], [40], ANN [60], [61], RF [38], [39], HMM [11] and Decision Trees [29], which have been utilized different DNA encoding

methods. However, existence of enormous pseudo (false) splice sites compared to real (true) sites in genome necessitates developing methods to improve prediction accuracy [2]. This Thesis proposed several new methods with improved accuracy for splice site prediction.

1.3.1 Novel DNA Encoding Methods using AdaBoost

In the class of ensemble learning, Adaptive Boosting (AdaBoost) is a powerful classifier that adopts a cleverer way of averaging weak classifiers by increasing the weight of misclassified samples at each iteration [62]. Due to its outstanding classification accuracy and absence of tuning parameters (except a number of iteration) [63], the AdaBoost algorithm has been utilized widely in biology. In [64], a novel method, named MirID, was proposed for pre-miRNA classification using the AdaBoost algorithm. In another study, Lu et al. presented a new approach for enhancer prediction based on the AdaBoost algorithm and shape features of chromatin modifications which outperformed previous methods significantly [62]. A cascade AdaBoost based learning procedure was adopted by Xie et al. to produce an effective promoter prediction method [65]. Recently, the AdaBoost was used for biomarker discovery of microarray data for cancer classification [66]. However, our literature search has failed to identify documented research related to splice site detection using AdaBoost.

In this thesis, we have proposed three new DNA encoding methods for feature extraction by combining several approaches that have already proven successful in determining pattern around splice sites. For given encoding methods, AdaBoost classifier outperformed SVM, NN, and RF significantly. Also, the proposed method exhibited outstanding performance compared to eleven current state-of-the-art methods and several existing tools using two well-known Human splice site datasets, namely HS3D and NN269 and an independent test dataset. The developed online prediction server (HSSAda) based on proposed approach, which is freely available at https://pashaei.shinyapps.io/hssada, is believed to help the biological community for easy detection of splice sites.

Part of this contribution has been published in:

Pashaei, E. and Aydin, N., (2017). "Prediction of Human splice sites using AdaBoost with efficient DNA encoding approaches", Frontiers of Information Technology & Electronic Engineering, (under review).

Pashaei, E. Ozen, M., and Aydin, N., (2016). "Splice Sites Prediction of Human Genome using AdaBoost", IEEE International Conference on Biomedical and Health Informatics (BHI 2016) Las Vegas, USA: 300-303.

Pashaei, E. Yilmaz, A. Ozen, M. and Aydin, N., (2016). "A Novel Method for Splice Sites Prediction Using Sequence Component and Hidden Markov Model", 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) Florida, USA: 3076 - 3079.

Pashaei, E. Yilmaz, A. Ozen, M. and Aydin, N., (2016). "Prediction of splice site using AdaBoost with a new sequence encoding approach", 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC) Budapest, Hungary: 3853-3858.

1.3.2 RF for Feature Selection and Classification

In biology, where structures are described by a large number of features as splice sites, the feature selection is an important step towards the classification task. It provides useful biological knowledge and allows for a faster and better classification. There are few specific works where feature selection techniques have been used in splice site prediction domain. Principle feature selection (PFA) is a multivariate filter method that has been employed by Maji [13] in Human splice site prediction. F-score feature ranking [1], [12] and estimated distribution algorithm (EDA) ranking methods [67] are two univariate filter methods that have been applied on human and plants splice sites, respectively. Also, the EDA has been utilized as a wrapper approach in [68] which has shown good performance in plant splice site prediction.

RF is among the most popular machine learning methods due to their relatively good performance. They also provide method for feature selection [69], [70], [71]. The random forest feature ranking (variable importance) has been used in the various domain such as integrated analysis of multiple data type [72], biomarker discovery [73] and multi-label classification [74]. In this thesis, we investigate the ability of RF feature ranking methods on the splice site prediction.

On the other hand, the SVM classifier is frequently used in prediction of splice sites due to its high performance. However, some parameters of SVM classifier such as penalty parameter, the kernel type, and kernel parameters, must be tuned. Parameter tuning can be time-consuming when there are multiple parameters involved in the training. So, one

should be cautious whether SVM is a suitable method for genome-wide splice sites prediction or not [11]. Keeping the above in view, we have combined RF as an efficient and fast classifier with three predefined encoding methods (MM1 [7], MM2 [13], MCM [10], [14], [61]) and compared their results with the SVM. We have also have evaluated the efficiency of proposed methods by making a comparison with some current methods. The experimental results show that the RF outperforms the SVM when the same Markovian encoding methods are used on both donor and acceptor datasets. Furthermore, the RF classifier performs much faster than the SVM classifier in detecting the splice sites, which make it suitable for use in the genome-wide project.

Part of this contribution has been published in:

Pashaei, E. Ozen, M., and Aydin, N., (2017). "Splice site identification in the human genome using random forest", Health and Technology, 7: 141-152.

Pashaei, E. Ozen, M., and Aydin, N., (2016). "Random Forest in Splice Site Prediction of Human Genome", XIV Mediterranean Conference on Medical and Biological Engineering and Computing Paphos, Cyprus: Springer International Publishing: 512-517.

1.3.3 A Novel DNA Encoding Method using SVM

We proposed another novel efficient DNA encoding methods based on MM2 using SVM, which outperform several well-known current methods. This contribution has been published in:

Pashaei, E. Yilmaz, A. and Aydin, N., (2016). "A Combined SVM and Markov Model approach for Splice Site Identification", 6th IEEE International Conference on Computer and Knowledge Engineering (ICCKE2016) Mashhad, Iran: 200-2004.

1.3.4 Organization

The rest of the thesis is organized as follows: We presented biological background relevant to our issue and related works in Chapter 1. We explained encoding methods, the machine learning algorithms used to predict splice sites and feature ranking methods in Chapter 2. In Chapter 3, we described our proposed methods. The experimental results were presented in Chapter 4. Chapter 5 provided conclusion.

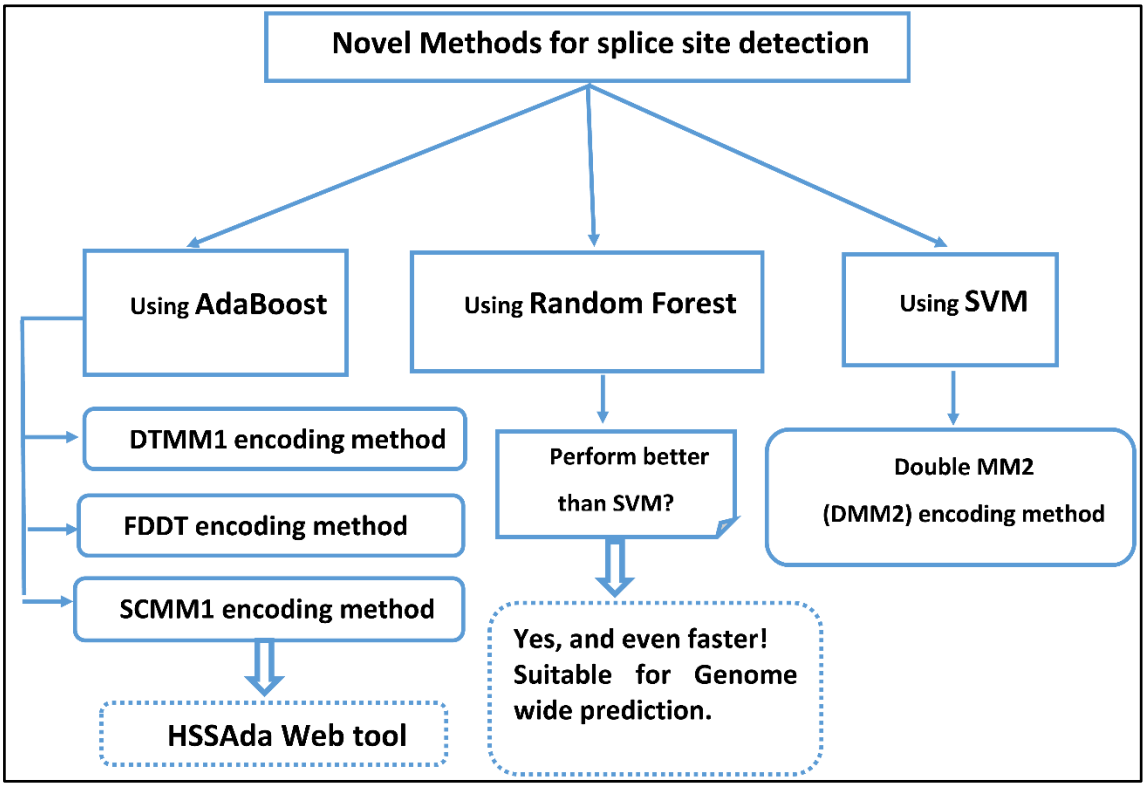

Figure 1.7 The overall structure of the contribution

CHAPTER 2

MATERIALS AND METHODS

2.1 Evaluation Datasets

We have utilized the well-known HS3D dataset [75] to evaluate the performance of the proposed methods. The HS3D consists of acceptor and donor splice sites. The sequences in both sites have the length of 140 nucleotides. The Acceptor segment is composed of 2880 true and 329374 false sequences, whereas the donor segment contains 2796 true and 271937 false sequences. The consensus dinucleotide AG has been placed at positions 69 and 70 for acceptor site, while consensus dinucleotide GT of donor sites has been located in positions 71 and 72. The performance of proposed methods is examined on both donor and acceptor sites separately. Two balanced and imbalanced datasets are extracted from both acceptor and donor sites. The balanced datasets (1:1) are constructed by selecting all of the true sequences and randomly choosing the same amount of sequences from false sites. However, the imbalanced datasets (1:10) are made by choosing all of the true sequences and randomly picking up 10 times more false splice sites than the true one.

To examine stability and reproducibility of the proposed methods, additional evaluation was performed on NN269 benchmark dataset [8]. This dataset has been split into the training set and test set for both acceptor and donor sites. The training dataset is made up of 1116 true acceptor sequences, 4672 false acceptor sequences, 1116 true donor sequences, and 4140 false donor sequences, while test dataset contains 208 true acceptor sequences, 881 false acceptor sequences, 208 true donor sequences, and 782 false donor sequences. The acceptor sequences have the length of 90 nucleotides, whereas the donor sequences have the length of 15 nucleotides. The consensus

dinucleotide AG has been placed at positions 69 and 70 for acceptor site, while consensus dinucleotide GT of donor sites has been located in positions 8 and 9. The training and test datasets are separate in the NN269.

We utilized an independent test set for comparing the performance of proposed methods with other tools. We prepared this dataset by using two different genes, namely AF102137.1 and M63962.1 downloaded from the Genebank [38]. Each gene has twenty true sites (without considering start and end of the gene) and many false sites. We constructed an imbalanced dataset by choosing all the true sites and randomly selected 4 times more false splice sites than the true one from both of the genes. Hence, the dataset consists of 40 confirmed true splice sites and 160 false splice sites for both donor and acceptor sites. The length and position of the dinucleotides are varied due to the differences in the format of the inputs of the tools for each of the donor and acceptor splice sites.

2.2 Feature Extraction

The DNA sequences are composed of four bases which are adenine (A), cytosine (C), guanine (G) and thymine (T). The DNA sequences are normally in string format. They must be converted to numerical feature vectors before feeding to classifiers. Therefore, it is necessary to employ DNA encoding methods to capture information from the DNA sequences.

2.2.1 Sparse Encoding Method

There are many DNA encoding methods in splice site prediction domain. The simplest encoding method allocates two binary digits to each nucleotide (A=00, C=01, G=10, T=11). However, it cannot reveal characteristics of DNA sequences. Another simple encoding method that assigns four binary digits to each nucleotide (A=0001, C=0010, G=0100, T=1000) is called sparse encoding method. The probability of natural mutation of DNA sequences cannot be considered using this approach [40].

2.2.2 Position Weighted Matrix Model

The position weighted matrix model (PWM), also called position specific weighted matrix (PSWM), is one of the well-known methods in bioinformatics for discovering

motifs in nucleotide sequences [76]. There is a matrix element for all possible nucleotides (bases) at every position. Each element in PWM model, m_{ij}, is computed as

$$m_{ij} = \log_2\left(\frac{P_{ij}}{P_i}\right) \tag{2.1}$$

where P_i is the background frequency of nucleotide i (0.25 for all bases in below example) and P_{ij} is the position-specific nucleotide frequency for nucleotide i at position j. Figure 2.1 shows an example of calculating PWM encoding methods for several aligned sequences.

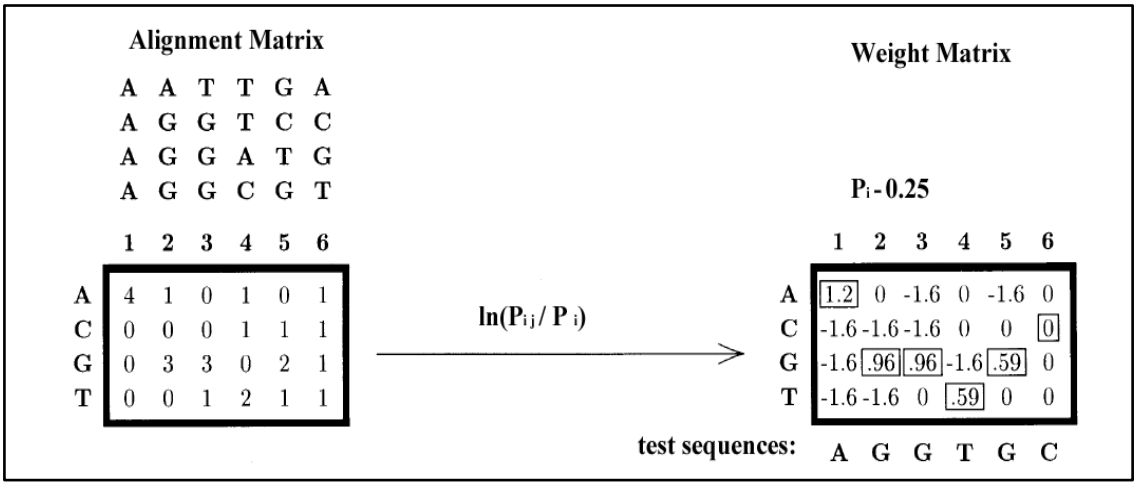

Figure 2.1 Example of the PWM model on several DNA aligned sequences [77]

2.2.3 Markovian Encoding Methods

The Markov model describes a sequence of possible states, in which the probability of each state depends only on the preceding states. In this thesis, MM1 encoding [1], [7], [12], MM2 encoding [13], and MCM [10], [14] encoding have been used.

The MM1 encoding method [1], [7], [12] is one of the well-known approaches which calculates the value of bases according to preceding base. The position specific probabilistic parameters are used to reveal the correlation between bases (nucleotides). Consider a sequence $(s_1, s_2, \dots, s_n)$ of length n. The nucleotide s_i is a realization of the i^{th} state variable in Markov chain. Each state is characterized by a position-specific probability parameter. The set of parameters in first order Markov model and second order Markov model are $\{P(s_i|s_{i-1})\}$ and $\{P(s_i|s_{i-1}, s_{i-2})\}$, respectively. The estimation of the model parameters is calculated by (2.2).

$$P(s_i|s_{i-1}, \dots, s_{i-k}) = \frac{N(s_{i-k},\dots,s_i)}{N(s_{i-k},\dots,s_{i-1})} \tag{2.2}$$

where k denotes the order of Markov model and $N(s_{i-k}, \dots, s_i)$ shows the occurrence number of $(s_{i-k}, \dots, s_i)$. In this study, $k = 1$ and $k = 2$ have been chosen for MM1 and MM2. As it is mentioned in [7], to create Markov model only true splice site sequences are considered. Figure 2.2 provided a small example of the MM1 methods in DNA sequence.

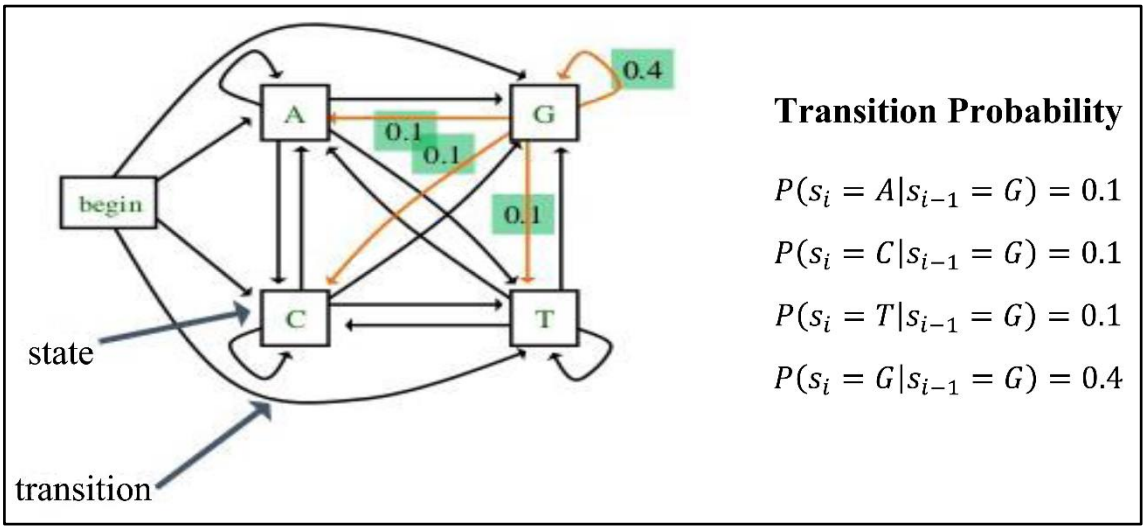

Figure 2.2 An example of first order Markov model

In the DM-SVM [12], its encoding part contains MM1 encoding method that extracts associated features by considering both true and false sites and produce $2(l-1)$ features, where l is the length of the sequence. However, we only take into account true splice sites to create the Markov model and subsequently $(l-1)$ feature is produced. The deficiency of MM1 encoding method is that the whole features of the splice site are not revealed by only considering the correlation between adjoining nucleotides.

The MCM was earlier used by Lio in [10] and again was employed recently in [14], [61]. This encoding method utilizes both MM1 and MM2 encoding methods. Each sequence is broken down into three parts: signal segment (S^S), upstream segment (S^U), and downstream segment (S^D), as shown in Figure 2.3. The signal segment is encoded by MM1 and the model is denoted by M_S. The upstream segments and downstream segments are encoded using MM2 and denoted by M_U and M_D, respectively. We also define a false model M_F to characterize the signal segment for false splice sites. The final model is a combination of them, that is (M_U, M_S, M_F , M_D).

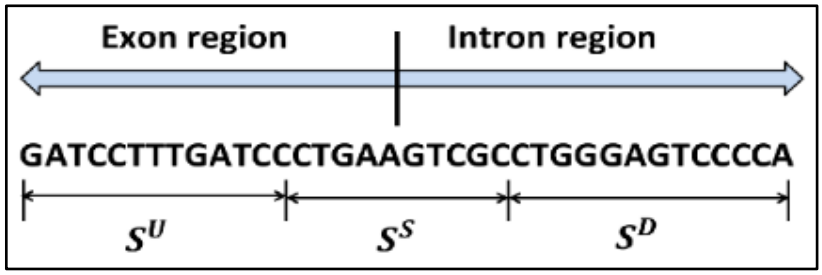

Figure 2.3 Model description of MCM encoding method [61]

We have set l_U=30, l_S=47, and l_D= 63 bp for donor sites, l_U= 48, l_S=21, and l_D=69bp for acceptor site in the HS3D dataset, while we have adjusted l_U=3, l_S=9, and l_D= 3bp for donor sites and l_U= 52, l_S=19, and l_D=19bp for acceptor site in the NN269 dataset.

2.2.4 Frequency Difference based Encoding Methods

The MN-FDTF encoding method [37], [38] allocates an integer number to each DNA base (A-1, T-2, G-3, and C-4). Then, two position weighted matrices are calculated using the true (TSS) and false splice sites (FSS) by counting the frequency of each nucleotide at a given position using (2.3).

$$M_{ij} = \frac{1}{n} \sum_{t=1}^{n} O_i\left(N_{tj}\right) \ , i = A, T, C, G \tag{2.3}$$

$$O_i(x) = \begin{cases} 1, i = x \\ 0, else \end{cases} \ ; j = 1,2, \dots l$$

where n is the number of sequences in the TSS and FSS, l is the length of a sequence and $O_i\left(N_{tj}\right) \in \{A, C, G, T\}$. The final encoding matrix is obtained by calculating difference between these two weighted matrices. The same process is repeated for PN-FDTF encoding method [37] with this difference that there are 16 integer numbers for assigning to each dinucleotide (AA-1, AG-2, AC-3, AT-4, GA-5, GG-6, GC-7, CA-9, CG-10, CC-11, CT-12, TA-13, TG-14, TC-15 and TT-16). Figure 2.4 provides diagrammatic representation of frequency encoding for preparation of training and test datasets.

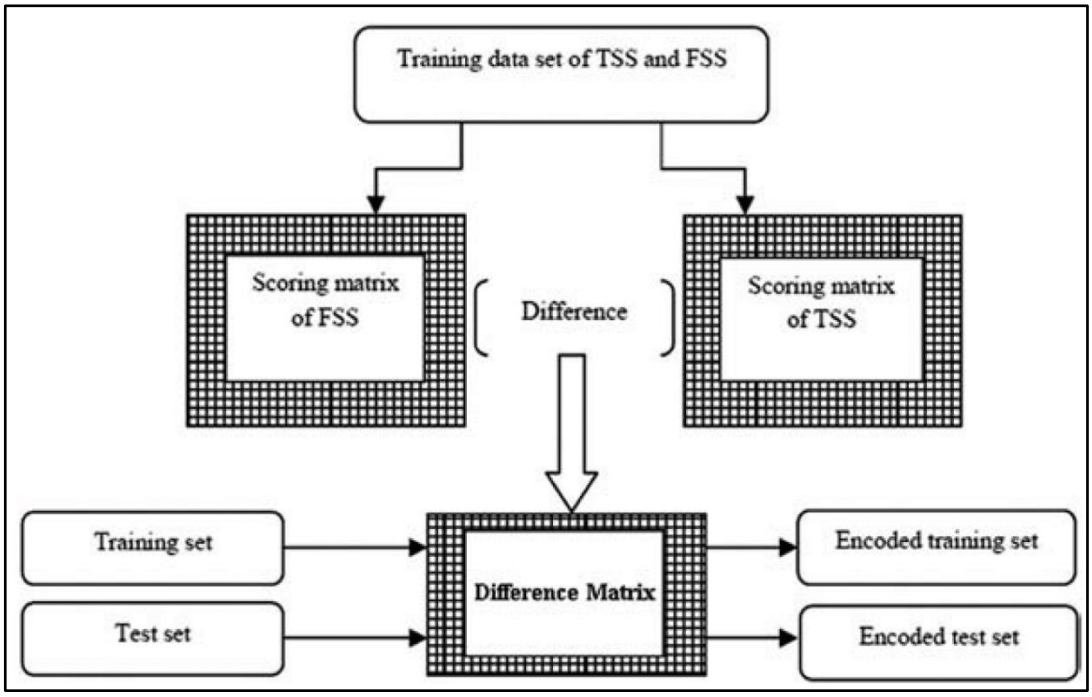

Figure 2.4 Overview of the process of the FDTD encoding methods [78]

2.2.5 Position Independent-Component based Encoding Method

The sequence position independent component (SC) encoding method [2], [27], [78], also known as K-mer, breaks the DNA sequences into two parts which are called upstream and downstream. Then, the probability of a string of bases $\alpha_1\alpha_2 \dots \alpha_k$ that is appeared in the sequences is calculated by (2-4).

$$P(\alpha_1\alpha_2 \dots \alpha_k) = \frac{n(\alpha_1\alpha_2 \dots \alpha_k)}{(l-k+1)} \quad (2.4)$$

where l is the length of the sequence, each α_i is one kind of DNA bases and $k \in \{1,2,3,4\}$. Let $n(\alpha_1\alpha_2 \dots \alpha_k)$ be the number of times the string $(\alpha_1\alpha_2 \dots \alpha_k)$ occurs in the sequence, by shifting one nucleotide position at a time. There are 4^k features to be extracted for each sequence [2]. The algorithm of the SC encoding has been given in Algorithm 2.1.

The reason behind of choosing $k <= 4$ is that although the protein-coding potential of transcripts is assessed by considering triplet bases, the regulatory element motifs are made up 6 bases. The experimental results in [2] indicate that considering the amount of k up to 4 produces satisfactory results.

Algorithm 2.1 SC based feature extraction

input: sequences $(S_1, S_2, \dots, S_N)$

output: feature matrix SC^t_{N*680}

begin

for $i = 1$ to N **do**

Separate each sequence S_i to two parts upstream (u) and downstream (d)

The following sequence component features are extracted for each part by using equation (3) :

Feature vectors SC for $k = 1,\ SC^u_{i*4}, SC^d_{i*4}$

Feature vectors SC for $k = 2,\ SC^u_{i*16}, SC^d_{i*16}$

Feature vectors SC for $k = 3,\ SC^u_{i*64}, SC^d_{i*64}$

Feature vectors SC for $k = 4,\ SC^u_{i*256}, SC^d_{i*256}$

end for

Merge all of them: $SC^t_{N*680} = \sum_{k=1}^{4} SC^u_{N*4^k} + \sum_{k=1}^{4} SC^d_{N*4^k}$

end

2.2.6 Distribution of Triple Nucleotide Encoding method

The distribution of triple nucleotide (DT) encoding method has been explained in detailed in [12], [41]. The aim of this method is to find the behavior of candidate splice sites in triplet format to help their recognition. There are 64 triplet bases for a DNA sequence. The pseudo-code of DT encoding method has been exhibited in Algorithm 2.2.

Algorithm 2.2 DT based feature extraction

input: sequences $(S_1, S_2, \dots, S_N)$

output: feature matrix SI^t_{N*128}

begin

for $i = 1$ to N **do**

- Separate each sequence S_i to two parts upstream (u) and downstream (d)
- The sequence information features are extracted for each part by searching and locating each 64 distinct triplet bases

Algorithm 2.2 (cont'd)

for $j = 1$ to 64 **do**

- Calculate $\alpha_r = 1/p_r - p_{r-1}$, $1 \le r \le m$ where p_r is the location of the r^{th} occurrence of triplet j, $p_0 = 0$ and m stands for the number of occurrences of triplet j
- Calculate partial sum of α_r: $\beta_j = \sum_{r=1}^{j} \alpha_r$, $1 \le j \le m$
- Calculate the Pseudo-Entropy (PE) of discrete probability distribution using $PE(q_1, q_2, \dots, q_m) = \sum_{k=1}^{m} q_k e^{1-q_k}$, where $q_k = \beta_k / \sum_{k=1}^{m} \beta_k$ and $\sum_{k=1}^{m} q_k = 1$

end for

Merge upstream and downstream of each sequence: $SI_{i*128} = PE_{u*64} + PE_{d*64}$

end for

end

2.3 Feature Selection

While feature extraction methods transform the DNA sequences into features, which will be the input of machine learning methods, the feature selection methods remove all the redundant features with the aim of increasing classification accuracy and reducing computational complexity. Feature selection techniques by considering the method's output can be divided into two groups; wrapper methods and filter methods [79], [80]. The wrapper methods pick up the feature subset based on classifiers performance. However, the filter methods assess the relevance of features via univariate statistical criteria instead of cross-validation performance. So, the wrapper methods give better performance result than filter methods due to taking into account features dependencies and directly interacting with the classifier. However, they are computationally more expensive than filter approaches [80]. On the other hand, the filter methods are known as the fast, rapidly scalable and efficient feature selection approaches in bioinformatics [79], [80]. There are two types of filter methods, univariate and multivariate methods. Most filter methods in the literature are univariate [79]. Multivariate filter methods can find relationships among the features, whereas univariate methods consider each feature individually. Therefore, multivariate filter methods can not disclose mutual information between features [81]. There are many various wrapper and filter approaches in the literature. Particle swarm optimization (PSO), Genetic algorithms (GA), sequential

forward and backward selection are some examples of the wrapper approach, while chi-square, correlation coefficient, Fisher score (F-score) feature ranking are some examples of filter approaches.

2.3.1 Fisher Score Feature Ranking Method

Feature selection has been performed in splice site prediction domain mostly through F-score ranking method [1], [12]. The feature ranking methods typically assign a weight to each feature and rank them accordingly. Then informative features can be selected and low-scoring features are removed. F-score is a simple univariate filter approach, which is used for ranking features according to their discriminative powers. Given training instance, F-score ranking method is defined as the following:

$$F(x_j) = \left| \frac{\bar{x}_j^{(+)} - \bar{x}_j^{(-)}}{\sigma_j^{+} - \sigma_j^{+}} \right| \tag{2.5}$$

Given a dataset X_{N*H} with N sequences and H features, we calculate Fisher score for each feature x_j, $j = 1,2,\dots,H$, by computing mean and variance of the both positive and negative class labels of the associated feature. The high F-score value of an attribute demonstrates that this attribute has more discriminative power [36].

When the F-score of each feature was computed, we have to find an optimal threshold for choosing a subset of features whose discriminative powers are the highest. In this study, after calculating F-score of each feature, the average F-score value of all features is considered as the threshold. Thus, the features whose F-scores are more than the threshold are selected and a new subset of features is constructed.

2.3.2 Random Forest Feature Ranking Method

Ranking of variables can be obtained by utilizing the mechanism of random forest. Each tree in the random forest is constructed on 2/3 of the training data which are drawn randomly with replacement (bootstrap). The split in each node of the trees is selected from a subset of variables (features). After building trees of the forest, each tree is tested on the 1/3 of the samples which have not been selected for bootstrap. These samples are called the Out-Of-Bag (OOB) instances and error of predictive performance of them is shown with $Err(OOB)$. The OOB is used for ranking variables by permuting each variable (j) one-by-one in OOB dataset of all the trees and calculating error of

predictive performance of the permuted version of OOB data (Err_j)). Subtraction of these errors is calculated at the next step. Ultimately, the average error of subtraction results and associated variances are measured. Algorithm 2.3 explains steps of calculating the ranking of the features using RF clearly. The "FSelector" R package has been used for implementation of RF feature ranking method. More detailed explanation on RF can be found in [74], [82].

Algorithm 2.3 RF feature ranking method

---

input: training dataset D_{N*M} , number of trees (k) in forest, size of feature subset (m) that is considered at each node during tree construction

output: Importance of each feature

begin

for $i = 1$ to k **do**

- Draw a bootstrap sample of size N from the training dataset.
- Grow a random-forest tree T_i to the $2/3$ of bootstrapped data
- Give the leftover $1/3$ of samples (called OOB) to the tree T_i, and calculate the error rate $Err(OBB)$

for $j = 1$ to M **do** // *for each feature* $j \in M$

- Permute the value of feature j randomly for the OOB samples
- Compute the error rate for permuted version of OBB samples Err_j using tree T_i
- Calculate $d_j = Err_j - Err(OBB)$

end For

end for

for $j = 1$ to M **do**

- Aggregate total error rate from all trees and calculate variance for each feature

$\hat{d} = \frac{1}{k}\sum_{i=1}^{k} d_i^j$ and $S_d = \frac{1}{k-1}\sum_{i=1}^{k}(d_i^j - \hat{d})^2$

- Calculate variable importance $v_{j=}\, \hat{d}/S_d$

end for

end

2.4 Classification Methods

2.4.1 AdaBoost Classifier

The AdaBoost algorithm is a machine learning meta-algorithm [62], which chooses a weak classifier (in this case decision tree) and continuously refines itself by rising the weights of the falsely classified samples at each iteration [83], which can dramatically improve the accuracy of the AdaBoost. Figure 2.5 illustrates how AdaBoost works and how a set of the weak classifier can be strong classifier $h_{fin}(x)$.

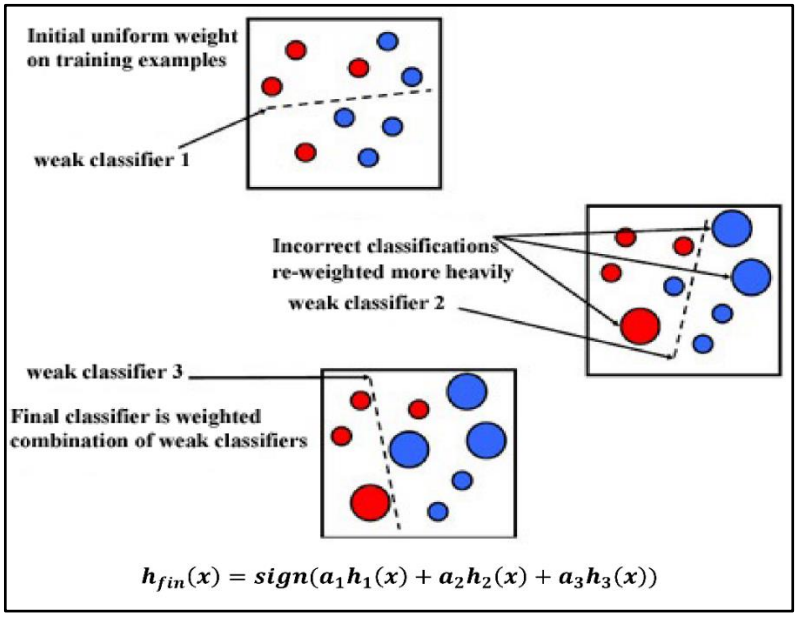

Figure 2.5 Illustration of AdaBoost classifier

The AdaBoost has some characteristics that make it so good for prediction task, such as its capability in utilizing many different classifiers, not being prone to overfitting, and easy implementation. There is various type of boosting algorithms which mainly differ in the ways that their errors represent and weights update. One of the well-known versions of the AdaBoost classifier is AdaBoost.M1 [84], [85], which has been employed in this study. The pseudo code for AdaBoost.M1 has been given in Algorithm 2.4.

The AdaBoost classifier requires only one parameter (number of iterations) to tune. The accuracy of the AdaBoost classifier increases by increasing the number of iterations. Consequently, training time (computational cost) also increases. We should note that there should be a stopping time for AdaBoost, i.e. the point from which test error starts to increase with the number of iterations [86].

Algorithm 2.4 Pseudo code of AdaBoost.M1

01 **input:** dataset of m samples $< (x_1, y_1), \dots, (x_m, y_m) >$ with labels $y_i \in Y = \{1, \dots, m\}$

02 **initialize:** $D_1(i) = 1/m$ such that $i = 1, \dots, T$

03 **begin**

04 **for** $i = 1$ to T **do**

05 Call weaklearn using distribution D_t

06 Get back a hypothesis $h_t: X \longrightarrow Y$

07 Calculate the error of h_t: $\epsilon_t = \sum_{i:h_t(x_i) \neq y_i} D_t(i)$

08 If $\epsilon_t > 1/2$, then set $T = t - 1$ and abort loop

09 Set $\beta_t = \epsilon_t/(1 - \epsilon_t)$

10 Update distribution D_t: $D_{t+1}(i)\ \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t & if\ h_t(x_i) = y_i \\ 1 & otherwise \end{cases}$ where Z_t is a normalization constant

11 **end for**

12 **output:** the final hypothesis: $h_{fin}(x) = argmax_{y \in Y}\ \sum_{t:h_t(x)=y} log\ \frac{1}{\beta_t}$

In order to prevent overfitting, early stopping has been emphasized by the statistics community [87]. For tuning the parameter of AdaBoost classifier, we have examined a definite range of numbers according to the length of sequences. For instance, the range has been set between 50 and 400 by step of 50 for the HS3D dataset. The maximum number of iteration has been considered 400 to achieve the model that not only has satisfactory prediction accuracy but also has a lower time cost.

We compared the performance of the AdaBoost.M1 with SVM [88], NN [89] and RF [90], because these techniques have been widely used for prediction in bioinformatics [38].

2.4.2 SVM Classifier

The SVM classifier has excellent empirical performance in many domains [91] and frequently used for detecting the splice sites [40]. The underlying idea of SVM classifier is to transform the input vector into a high-dimension Hilbert space and seeking a separating maximum margin hyperplane in this place (See Figure 2.6)

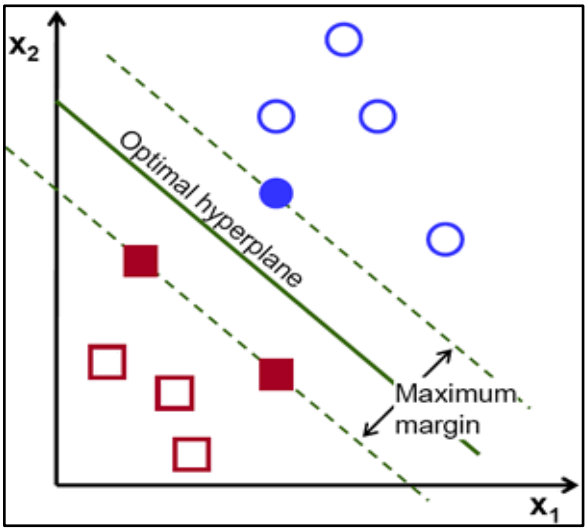

Figure 2.6 Maximum margin hyperplane and margins for an SVM trained with samples from two classes

The SVM classifier was trained with a radial basis function (RBF) kernel, while a grid-based search method was used to seek optimal parameters of the kernel (the soft margin parameter C and the Kernel parameter γ). Figure 2.7 explains the flowchart of the tuning process.

2.4.3 NN Classifier

The NN is another classifier that is frequently used in splice site prediction because of its ability to capture and represent complex input-output relationships [16]. The NN is an information processing model, which is composed of a large interconnected group of artificial neurons. The structure of an NN is represented as multiple layers of these neurons working in parallel to solve a specific problem [92].

In the NN model, three parameters which are “size”, “decay” and “maxit” should be tuned. The “size” parameters stand for the number of units in the hidden layers, while the “decay” and “maxit” parameters demonstrate weight degradation and maximum number of iteration. Again, a grid-based search approach was utilized for finding optimal values.

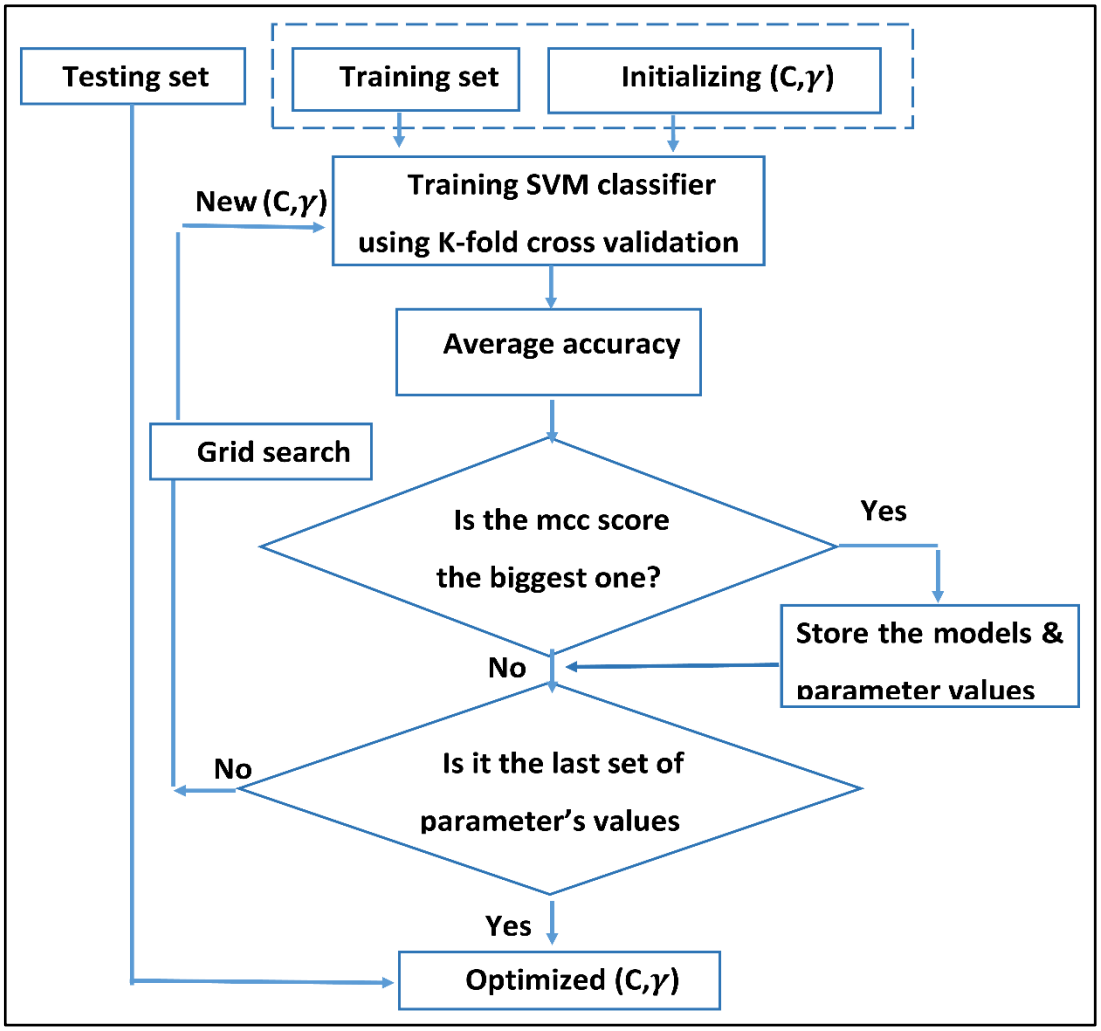

Figure 2.7 Flowchart of SVM tuning process

2.4.4 RF Classifier

The RF classifier, which has been introduced by Breiman in 2001[90], is an ensemble learning method based on decision tree which has been utilized in splice site prediction [38], [39] due to its high efficiency in prediction accuracy and time complexity. Each tree in the forest is trained by randomly selecting samples with replacement (bootstrap) from total samples of the original dataset. The rest of the samples are used as the test set. A single decision tree uses randomly m number of features from total M features in splitting each node (*mtry*). A random forest with k decision tree (*ntree*) repeats above procedure for each decision tree and final classification is obtained by the voting result of these k decision trees on testing data. Figure 2.8 shows RF framework using two trees, while Algorithm 2.5 describes the steps of Random Forest classifier.

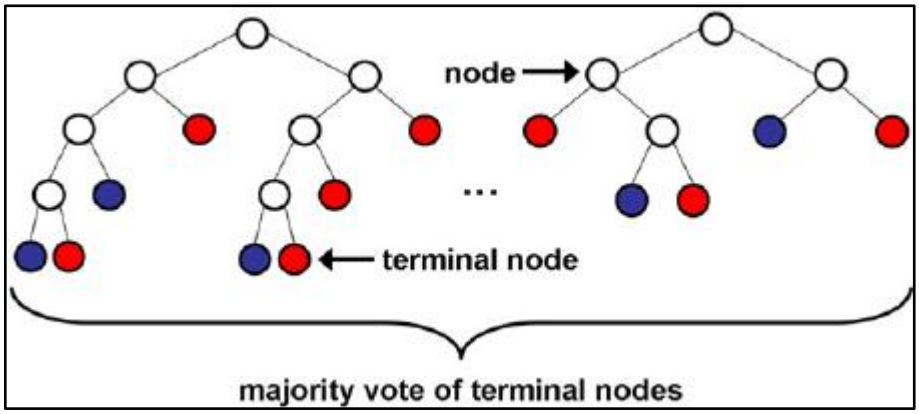

Figure 2.8 Illustration of a random forest [93]

Algorithm 2.5 RF classifier framework

---

input: training dataset D_{N*M} , number of trees (k) in forest, size of feature subset (m) that is considered at each node during tree construction

begin

for $i = 1$ to k **do**

- Draw a bootstrap sample of size N from the training dataset.
- Grow a random-forest tree T_i to the 2/3 of bootstrapped data, by recursively repeating the following steps for each terminal node of the tree until the minimum node size n_{min} is reached.
  - Select m features at random from total M features.
  - Pick the best feature/split-point among the m.
  - Split the node into two daughter nodes.

end for

output the ensemble of trees $\{T_i\}_1^k$.

end

To make a prediction at the new point x:

$\hat{C}_{RF}^k(x) = majority\ vote\ \{\hat{C}_i(x)\}_1^k$, let $\hat{C}_i(x)$ be the class prediction of the ith tree in RF.

We have implemented Random Forest algorithm using "Random Forest" package in R software. The Random Forest has two parameters for tuning namely "*mtry*" and "*ntree*". They are a number of features to choose at each node for splitting and number of trees to be grown in the forest respectively. In this study, "*mtry*" is equal to $\sqrt{M}$, while

"*ntree*" is equal to 500 (default value) on the HS3D dataset. The value of "*ntree*" has been set to 530 for the NN269 dataset.

2.5 Statistical Comparison

When we compare performances of classifiers, it is important to determine whether the observed difference in their performance is statistically significant or simply due to chance. The Mann-Whitney U test was utilized to assess the significance of differences in classification performance of proposed and existing approaches. The null hypothesis of this test is that there are no differences between performances of AdaBoost and other classifiers, while a significance level of α=0.01 was considered for this test. F-measure criteria were used to perform this test.

2.6 Implementation

The "adabag" R package was used for implementing the AdaBoost.M1 classifier, while the "e1071", "Random Forest", and "nnet" packages of R were employed for implementation of SVM, RF, and NN, respectively. Also, the Mann-Whitney U-test was performed by using "stats" R package. All of the programs were written in R 3.3.3 and run on the windows 7 operation system on 5 core 2.40 GHz CPU and 8 GB main memory.

2.7 Evaluation Criteria

This study has utilized several criteria to measure the performance of prediction methods. They are sensitivity (S_n), specificity (S_p), global accuracy (Q^9), Matthew's correlation coefficients (Mcc), area under ROC curve (AUC), area under precision-recall curve ($auPRC$) and F-measure.

$$S_n = TP/(TP + FN) \tag{2.6}$$

$$S_p = TN/(TN + FP) \tag{2.7}$$

$$Q^9 = (1 + q^9)/2 \tag{2.8}$$

$$q^9 = \begin{cases} \dfrac{(TN - FP)}{(TN + FP)} & if \;\; TP + FN = 0 \\ \dfrac{(TP - FN)}{(TP + FN)} & if \;\; TN + FP = 0 \\ 1 - \sqrt{2\left[\left(\dfrac{FN}{TP + FN}\right)^2 + \left(\dfrac{FP}{TN + FP}\right)^2\right]} & if\ TP + FN \neq 0 \\ & and\ TN + FP \neq 0 \end{cases}$$

$$Mcc = \frac{(TP*TN)-(FP*FN)}{\sqrt{(TP*FN)*(TN*FP)*(TP*FP)*(TN*FN)}} \tag{2.9}$$

$$F - measure = 2 * \frac{TP/(TP+FP)* S_n}{TP/(TP+FP)+ S_n} \tag{2.10}$$

where TP, FN, TN, and FP represent the number of true positives, false negatives, true negatives and false positives, respectively. Due to the existence of the false splice sites than the true ones, we confronted with imbalanced classes in splice site prediction problem. So, the need of measures that are independent of class distribution is essential and we have fulfilled it by considering Q^9, Mcc, and $F - measure$ criteria.

On the other hand, the AUC is commonly used when evaluating binary decision problems. It exhibits that how the number of the correctly classified positive samples changes with the number of incorrectly classified negative samples [94]. However, when dealing with highly imbalanced datasets, the $auPRC$ have been cited as an alternative to AUC [3], [94].

2.8 Cross-Validation Design

To evaluate the effectiveness of classification, a 10-fold cross validation has been used on the HS3D dataset. The dataset is separated into 10 equal size segments (folds). The 9 out of 10 folds are used for training, whereas remaining fold is used for testing. This process is repeated 10 times by choosing different folds as the test set and the total average is reported. To increase the reliability of the evaluation, the whole process is replicated 5 times by breaking the dataset into 10 different folds and the overall average is calculated. However, the process of validation on imbalanced datasets of HS3D are slightly different due to the large inequality between the number of true sites and false sites. It causes that the evaluating indicators tend to be biased towards the majority class [95]. In order to solve the problem of imbalanced classification, the under-sampling technique [96], [97] has been employed. We separate the dataset into 10 equal sizes in

such a way that the ratio between true sites and false sites (1:10) are the same in each constructed folds. Then, the training dataset (the combination of 9 out of 10 fold) is modified so that the same portion of the number of false sites versus the number of true sites are chosen in each iteration of cross-validation for training the classifier (random undersampling). However, the test set (remaining fold) was not modified and remained imbalanced.

Since the training data and test data are separate at the NN269, we do not need any cross-validation approach in its evaluation process. However, in order to tune parameters of SVM, we divided training dataset into 10 equally sized data fold. Each fold contains the same proportion of true versus false sequences. For each parameter combination, we used 9 out of 10 folds and evaluated the methods on the remaining fold. We selected the model with the highest average of auPRC on 10 evaluation sets. Then this best model was trained on the complete training dataset. The ultimate evaluation was performed on the corresponding independent test sets. According to [98], when the binary classifier on the imbalanced dataset is evaluated, the auPRC is more informative than AUC. So, we focused on the auPRC measure for model selection of SVM.

2.9 Online Predictor Server

Based on the proposed approach, a web server, named HSSAda, was developed using "Shiny" R package in RStudio to help the biological community in predicting splice sites. The developed R-code was run in the background, while the model was trained with the HS3D dataset. To submit the sequence(s), the user can either paste the sequence(s) in a text area or upload FASTA file. The results are displayed in terms of probabilities and labels, which "yes" indicates true splice site and "no" demonstrates false one. We did not consider any threshold for probabilities, whereas we adjusted threshold value of 0.4 and 3 for the NNsplice and MaxEntScan, respectively.

CHAPTER 3

PROPOSED APPROACHES

3.1 Novel Encoding Methods using AdaBoost

We presented three novel DNA encoding methods based on a hybrid of the Markovian model (MM1), distribution of tri-nucleotide, frequency difference of mono and di-nucleotide, and position-independent component features. To predict splice sites, four well-known classifiers (Support Vector Machine, AdaBoost, Random Forest, and Artificial Neural Network) have been used. The performance of the proposed methods was evaluated on two most popular publicly available Human splice site datasets, HS3D and NN269. The experimental results demonstrated that the AdaBoost outperformed all the considered classifiers using proposed encoding methods.

3.1.1 The DTMM1 Encoding Method

This encoding is hybrid of the DT and MM1 encoding methods. As mentioned before, the main difference between the proposed method and Wei's study [12] is the MM1 encoding part, which is created by considering only true splice sites sequences. After calculating MM1 and DT for all the DNA sequences, we use the F-score ranking method to select the discriminative features. We merge final MM1 vectors and DT vectors as the input of classifier. The steps of the proposed method have been described in Algorithm 3.1.

Algorithm 3.1**.** The steps of the DTMM1 approach

01 **input:** The candidate splice site sequences, $(S_1, S_2, \dots, S_N)$, length of sequence , l

02 **output:** Labels of unknown sequences

03 **encoding steps:**

04 • Using distribution of triple nucleotide encoding to construct A_{N*128} matrix

05 • Using First order hidden Markov model encoding to construct B_{N*l-1} matrix

06 • Using F-score feature ranking method to choose subset of features for both of them A_{N*a}, B_{N*b}

07 • Merge both of them respectively (A_{N*a} , B_{N*b}) to produce training set

08 **classification:** Feed them to classifiers to make prediction

3.1.2 The FDDT Encoding Method

This encoding method is called frequency difference distribution of triple nucleotide (FDDT) due to utilizing the MN-FDTF, the PN-FDTF and the DT encoding methods to extract features. Before merging the extracted features, we apply F-score to the MN-FDTF features and determine the position of the features whose F-score are more than the average threshold. By taking these positions into account, a new contiguous position vector is constructed. In other words, it contained consensus sites AG for acceptor and GT for Donor sites besides the selected position by F-score. The new position vector is used to select features from the MN-FDTF. The same process is repeated for the PN-FDTF. Finally, we merge them to provide the input to the classifiers. The pseudo-code of the FDDT algorithm has been given in the Algorithm 3.2.

Algorithm 3.2 The steps of the FDDT approach

01 **input:** The candidate splice site sequences, $(S_1, S_2, \dots, S_N)$, length of sequence , l

02 **output:** Labels of unknown sequences

03 **encoding steps:**

04 • Using distribution of triple nucleotide (DT) encoding to construct A_{N*128} matrix.

05 • Using MN-FDTF encoding to construct B_{N*l} matrix

06 • Using PN-FDTF encoding to construct C_{N*l-1} matrix

Algorithm 3.2 (cont'd)

- Using F-score feature ranking method to choose subset of features (a) for matrix B_{N*a} and determine position of chosen features.
- Construct a new contiguous vector of position by adding position of AG for acceptor and position of GT for donor site among the selected position from previous step, B_{N*a+2}.
- The same process is used to choose subset of features for matrix C, C_{N*a+2}.
- Merge all of them respectively (B_{N*a+2} , C_{N*a+2} , $A_{N*(b\ or\ 128)}$) to produce training set.

classification: Feed them to classifiers to make prediction

For more illustration, let us consider donor splice site of the NN269 dataset that is composed of 15 nucleotides. After calculating the mono-nucleotide and the pairwise nucleotides, the F-Score Feature ranking method is applied to them. Figure 3.1 shows the F-score values of each feature using the MN encoding. The amount of threshold is 0.17253 (calculated by averaging all the F-score values). By considering the threshold, features whose F-scores are above the 0.17 are chosen. These features have been placed at the position sixth, seventh, tenth, eleventh and twelfth. For constructing a contiguous vector of the position we add features at the positions eighth and ninth to the previous vector. So the new vector includes the features in the (6, 7, 8, 9, 10, 11, 12) positions. Features of the pair-wise nucleotides are selected by using the same process.

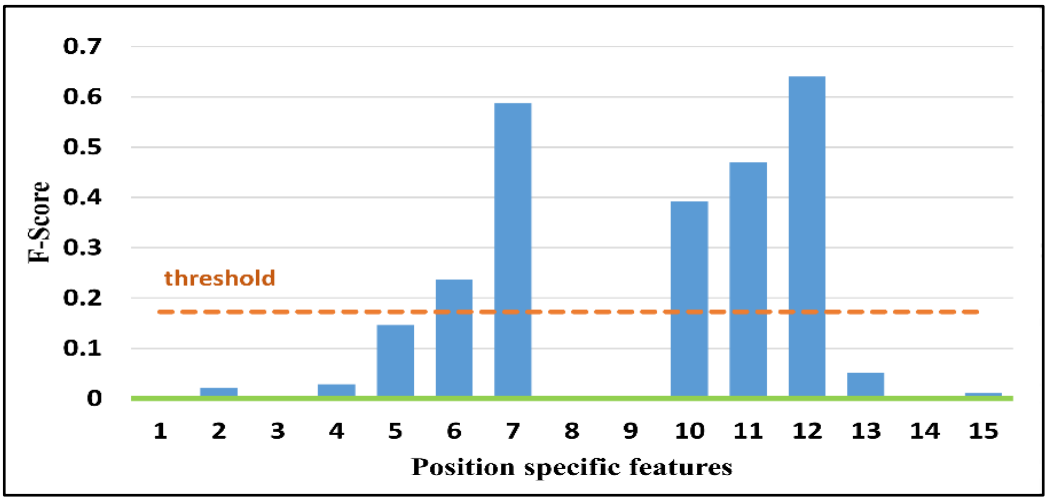

Figure 3.1 The F-score values on the features which were obtain by MN-encoding method on NN269 Donor sites

3.1.3 The SCMM1 Encoding Method

In this method sequence component is calculated for all of the sequences and the F-score feature ranking is applied to select the informative features. Then, the MM1 features are extracted and again the F-score feature ranking method is applied for choosing the contiguous features. At the end, the SC and MM1 features are combined to provide the input to the classifiers. The pseudo-code of the SCMM1 algorithm is illustrated in Algorithm 3.3.

Algorithm 3.3 The steps of the SCMM1 approach

---

input: The candidate splice site sequences, $(S_1, S_2, \dots, S_N)$, length of sequence , l

output: Labels of unknown sequences

encoding steps:

- Using Sequence component encoding to construct A_{N*680} matrix
- Using MM1 encoding to construct B_{N*l-1} matrix
- Using F-score feature ranking method to choose subset of features (a) for matrix A, A_{N*a}
- Again, using F-score feature ranking method to choose subset of features (b) for matrix B, B_{N*b} and also add AG for acceptor and GT for donor site among the selected features, B_{N*b+2}
- Merge both of them respectively (A_{N*a+2} , B_{N*b}) to produce training set

classification: Feed them to classifiers to make prediction

3.2 Performance of RF on Markovian Encoding Models

This part is concerned with RF for feature selection [70] and classification in splice site prediction domain. The performance of RF ranking method has been compared with F-score feature ranking [36] by using the learning curve concept. Liu [99] and Kocev [74] have remarked on the use of learning curves to show the effect of adding features when a list of ordered features is provided. We have investigated their effect on HS3D datasets with the goal of using a small number of features to achieve better classification performance.

3.2.1 RF as Feature Ranking

We proposed the following two-step procedure (See Algorithm 3.4) for investigating RF feature ranking approach in Human splice site detection. At the first step, we have applied RF feature ranking method to train dataset. As the consequence, a value is assigned to each feature which demonstrates the importance of the each feature in classification accuracy. Then, we sorted them according to their values decreasingly. At the second step, we evaluated the ranking by performing a stepwise feature subset evaluation, which is used to provide the learning curve. For this purpose, we selected the top-k ranked features from the ordered variables. Then, we evaluated the performance of the classifier on chosen subset feature and constructed forward feature addition curve (FFA).

Algorithm 3.4 Steps of providing forward feature addition curve using Random Forest

---

input: The provided training data D_{N*M}, number of total features M

output: Forward feature addition curve (FFA)

begin

Compute the RF score of importance for all the feature. $R = \{I_1, I_2, \dots, I_M\}$ is the vector of obtained feature ranking.

Order the features in decreasing order of importance

for $i = 1$ **to** 10 **do**

- select k-top ranked feature from R and accordingly carry out feature selection on training set, $k = (i * 10 * M)/100$
- Apply SVM on the training set D_{N*k} to learn the prediction model
- Use the model to make prediction on the test set with the chosen k features(calculate Q^9)
- Return Q^9 measurement for drawing FFA curve

end for

end

3.2.2 RF as Classifier

Splice site is subdivided into two separate classification problems: acceptor splice site classification and donor splice site classification. We try to identify whether a candidate splice site is true splice site (positive) or not (negative) for both of the problems. So,

two different models are constructed for them to make a prediction. These models consist of two phases: feature extraction using encoding scheme and classification. The proposed methods MM1-RF, MM2-RF, and MCM-RF utilize Markovian encoding approaches MM1, MM2 and MCM to provide features and use RF for classification. The steps of models are outlined in Algorithm 3.5.

Algorithm 3.5 Steps of the proposed splice site prediction methods MM1-RF, MM2-RF and MCM-RF

---

input: The candidate splice site sequences, $\{S_1, S_2, \dots, S_N\}$

output: Labels of unknown sequences

begin

for $i = 1$ to N **do**

- Model S_i using one of the proposed Markovian encoding methods (MM1, MM2 or MCM). The Output is a vector of features, $F_i = (f_1, f_2, f_3, \dots)$

end for

Apply RF on the training set of the extracted features $\{F_1, F_2, \dots, F_N\}$ to learn the prediction model

Use the model to make prediction on the test sequences of splice sites

end

3.3 A novel Encoding Method using SVM

The proposed method consists of two steps. In the first step, we extract informative features from DNA sequences by employing DMM2 (Double MM2) encoding approach. Two matrices A_{N*l-2} and B_{N*l-2} are calculated by using TSS (M^T) and FSS (M^F), respectively. Then, the final feature matrix, $C_{N*2(l-2)}$ is constructed as the input of the classifier by combining A_{N*l-2} with B_{N*l-2} . In the second step, the feature vectors are fed to SVM for classifying the splice sites. Algorithm 3.6 explain steps of proposed method.

Algorithm 3.6 Proposed DMM2-SVM method

---

input: the candidate splice site sequences, $(S_1, S_2, \dots, S_N)$, length of sequence , l

output: labels of unknown sequences

steps:

Algorithm 3.6 (cont'd)

04 Calculate M^T model using True splice site and Compute feature vectors A^T_{N*l-2} using M^T

05 Calculate M^F model using False splice site Compute feature vectors B^F_{N*l-2} using M^F

06 Merge both of them $C_{N*2(l-2)} = (A^T_{N*l-2}, B^F_{N*l-2})$

07 Apply the SVM classifier on the training set to obtain the model and use the model to predict the splice sites on testing sequences.

CHAPTER 4

RESULTS

4.1 Experimental Results using AdaBoost

To evaluate the performance of our proposed methods, we utilized two popular datasets, namely HS3D and NN269. We assessed the efficiency of the proposed DNA encoding methods by considering four outstanding classifiers, namely SVM, NN, RF, and AdaBoost, to seek the most successful model. Then, we compared results of the chosen model with current state-of-the-art methods for both of the datasets. To examine the performance of the designed web tool (HSSAda), several well-known tools were used for comparison using an independent test dataset.

4.1.1 Evaluation on the HS3D Dataset

The 10-fold cross-validation was run on both acceptor and donor sites of balanced and imbalanced datasets using SVM, NN, RF, and AdaBoost classifiers. The results of the proposed methods have been shown in Table 4.1, Table 4.2, Table 4.3 and Table 4.4 for balanced and unbalanced acceptor and donor sites, respectively. Further, the performance comparisons of the models with twelve state-of-the-arts methods were reported in Table 4.5, Table 4.6, Table 4.7 and Table 4.8 for both balanced and unbalanced acceptor and donor sites, separately.

4.1.2 Optimum Value of Parameters

Figure 4.1 shows the performances of the three proposed encoding methods using the AdaBoost classifier for various iterations on both donor and acceptor sites of balanced and imbalanced datasets.

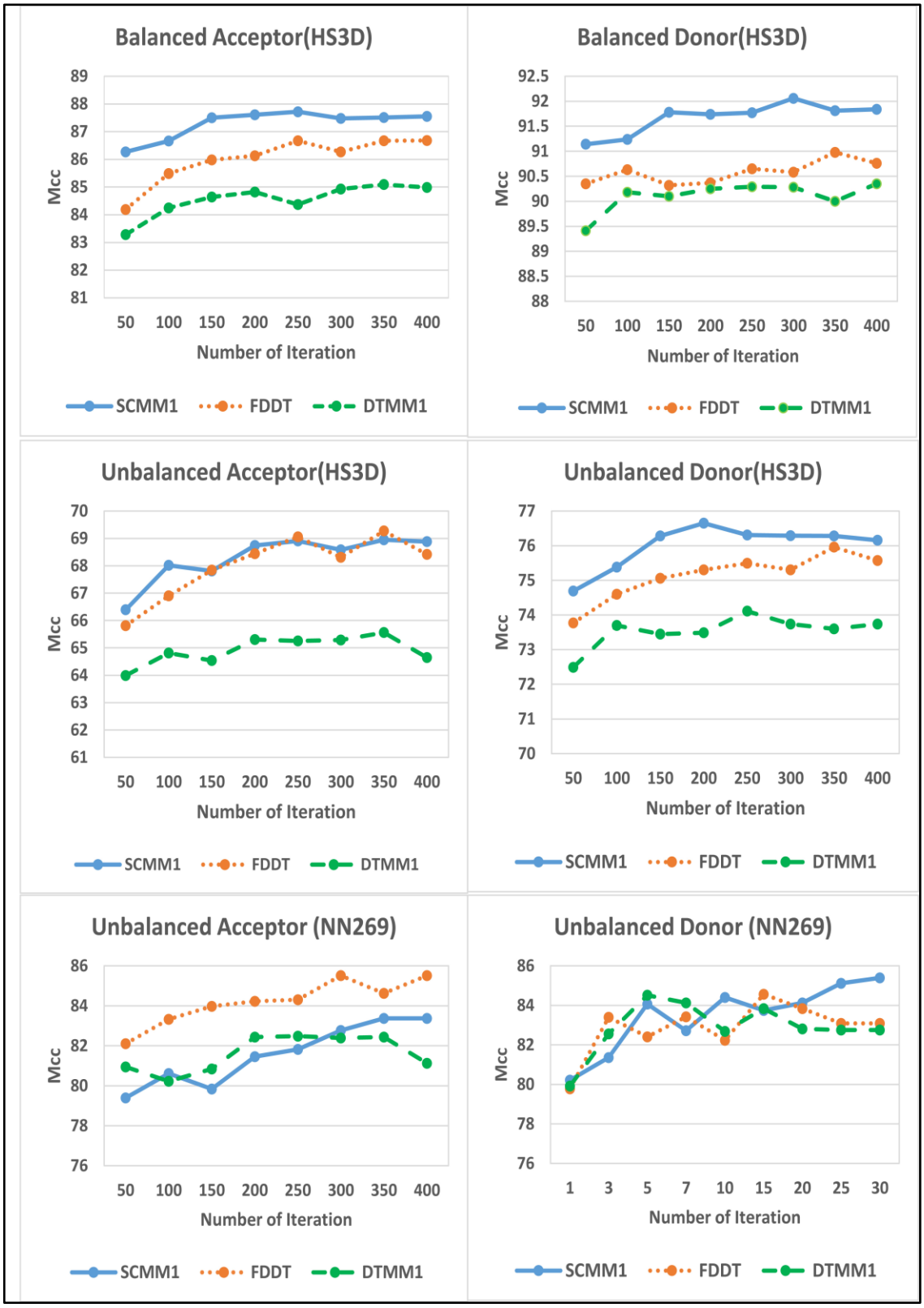

Figure 4.1 Evaluation of AdaBoost performance at different numbers of iteration on HS3D and NN269 datasets

The best result of the DTMM1, FDDT and SCMM1 encoding methods, using AdaBoost classifier, have been obtained in iteration 350, 250 and 250, respectively in balanced acceptor sites. According to the above order of the mentioned encoding methods, the high performance is seen in the iteration of 400, 350 and 300 for the balanced donor sites. By considering the imbalanced datasets, the value of optimal parameters in acceptor sites is the same and is equal to 350, whereas they are 250, 350 and 200 in donor site, respectively.

Table 4.1 Performance comparison of the SVM, RF, NN, and AdaBoost under three proposed encoding methods on predicting balanced Acceptor splice sites of HS3D datasets

Encoding Methods	**Candidate classifier**	**Balanced Acceptor sites**						**p-value (AdaBoost)**
		S_n	S_p	Q^9	Mcc	**AUC**	**F-measure**	
DTMM1	SVM	92.92	88.99	90.69	81.99	97.11	91.12	0.00194
		(1.60)	(0.88)	(0.76)	(1.75)	(0.42)	(0.89)	
	NN	92.15	89.24	90.53	81.44	96.70	90.82	0.000765
		(1.66)	(1.32)	(1.02)	(1.95)	(0.43)	(1.00)	
	RF	91.77	90.45	91.03	82.24	96.89	91.16	0.0113
		(1.75)	(1.23)	(1.05)	(2.14)	(0.25)	(1.10)	
	AdaBoost	**93.72**	**91.04**	**92.16**	**84.82**	**97.41**	**92.48**	--
		(1.70)	(1.35)	(0.74)	(1.52)	(0.39)	(0.78)	
FDDT	SVM	92.99	90.55	91.64	83.58	97.47	91.87	0.000155
		(1.31)	(0.76)	(0.54)	(1.21)	(0.45)	(0.63)	
	NN	92.74	91.00	91.78	83.77	97.33	91.94	0.000501
		(1.08)	(1.44)	(0.89)	(1.77)	(0.36)	(0.86)	
	RF	91.35	89.62	90.37	81.01	96.81	90.57	0.000154
		(1.13)	(1.56)	(0.51)	(0.98)	(0.28)	(0.44)	
	AdaBoost	**94.76**	**91.87**	**93.13**	**86.68**	**98.10**	**93.41**	--
		(0.69)	(0.94)	(0.65)	(1.33)	(0.37)	(0.66)	
SCMM1	SVM	93.51	91.21	92.24	84.75	97.74	92.45	0.000326
		(0.91)	(1.10)	(0.69)	(1.37)	(0.30)	(0.67)	
	NN	93.89	92.22	92.98	86.13	97.88	93.11	0.01261
		(0.91)	(0.91)	(0.66)	(1.32)	(0.26)	(0.66)	
	RF	91.88	90.34	91.04	82.24	97.00	91.17	0.000157
		(1.73)	(1.08)	(1.15)	(2.42)	(0.37)	(1.24)	
	AdaBoost	**95.07**	**92.60**	**93.64**	**87.72**	**98.14**	**93.91**	--
		(1.16)	(1.40)	(0.77)	(1.40)	(0.33)	(0.69)	

The value inside the brackets () are the standard errors
The reported p-values have been computed between AdaBoost and other classifiers using F-measure criteria

Table 4.2 Performance comparison of the SVM, RF, NN and AdaBoost under three proposed encoding methods on predicting unbalanced Acceptor splice sites of HS3D datasets

Encoding Methods	Candidate classifier	UnBalanced Acceptor sites						p-value (AdaBoost)
		S_n	S_p	Q^9	Mcc	AUC	F-measure	
DTMM1	SVM	93.02	89.72	91.19	62.23	97.08	62.94	0.00194
		(1.15)	(0.76)	(0.59)	(1.58)	(0.25)	(1.62)	
	NN	92.88	89.44	90.97	61.57	96.79	62.26	0.000669
		(1.14)	(0.61)	(0.51)	(1.22)	(0.18)	(1.25)	
	RF	92.12	90.42	91.14	63.08	96.88	64.01	0.01556
		(2.27)	(0.59)	(1.05)	(2.03)	(0.49)	(1.82)	
	AdaBoost	**93.96**	**91.04**	**92.35**	**65.56**	**97.42**	**66.30**	--
		(1.04)	(0.67)	(0.73)	(1.96)	(0.39)	(1.91)	
FDDT	SVM	93.06	90.83	91.84	64.53	97.35	65.38	0.000285
		(1.43)	(0.50)	(0.76)	(1.36)	(0.31)	(1.52)	
	NN	93.26	91.00	92.01	65.03	97.40	65.87	0.000506
		(1.38)	(0.58)	(0.65)	(1.50)	(0.24)	(1.47)	
	RF	91.94	89.84	90.80	61.76	96.77	62.66	0.000157
		(1.30)	(0.62)	(0.71)	(1.63)	(0.46)	(1.56)	
	AdaBoost	**95.24**	**92.30**	**93.54**	**69.27**	**98.16**	**69.98**	--
		(1.40)	(0.56)	(0.57)	(1.42)	(0.40)	(1.42)	
SCMM1	SVM	94.17	91.46	92.66	66.65	97.82	67.42	0.01906
		(1.31)	(0.78)	(0.83)	(2.18)	(0.34)	(2.15)	
	NN	94.17	91.70	92.78	67.19	97.96	67.98	0.01906
		(1.26)	(0.67)	(0.53)	(1.52)	(0.25)	(1.57)	
	RF	92.05	90.44	91.13	63.07	97.08	64.02	0.000156
		(1.95)	(0.69)	(0.76)	(1.29)	(0.42)	(1.29)	
	AdaBoost	**94.86**	**92.27**	**93.39**	**68.95**	**98.20**	**69.72**	--
		(1.25)	(0.61)	(0.55)	(1.54)	(0.27)	(1.56)	

The value inside the brackets () are the standard errors
The reported p-values have been computed between AdaBoost and other classifiers using F-measure criteria

4.1.3 Performance Comparison on different Classifiers

From Table 4.1, Table 4.2, Table 4.3, and Table 4.4 it can be seen clearly that the combination of proposed encoding methods with AdaBoost classifier performs nominally (that is, not necessarily statistically significantly) better than the composition of them with SVM, RF, and NN classifiers in all the terms for acceptor splice sites. The statistical comparison test with significance level of $\alpha = 0.01$ reveals that AdaBoost significantly outperforms other classifiers using proposed encoding methods, except

DTMM1-RF and SCMM1-NN, which perform the same. According to the results, we have chosen the SCMM1-AdaBoost method as our best method in all the terms for the balanced and the imbalanced acceptor sites. From Table 4.3 and Table 4.4, it is observed that for the balanced and the imbalanced donor sites the best performance again has been obtained by the SCMM1-AdaBoost in all the terms.

Table 4.3 Performance comparison of the SVM, RF, NN and AdaBoost under three proposed encoding methods on predicting balanced Donor splice sites of HS3D datasets

Encoding Methods	**Candidate classifier**	**Balanced Donor sites**						**p-value (AdaBoost)**
		S_n	S_p	Q^9	Mcc	**AUC**	**F-measure**	
DTMM1	SVM	94.46	91.85	92.96	86.35	97.82	93.24	0.000758
		(1.06)	(1.76)	(1.03)	(1.67)	(0.53)	(0.81)	
	NN	93.71	91.70	92.58	85.44	97.23	92.78	0.000379
		(1.14)	(1.64)	(1.03)	(1.87)	(0.58)	(0.91)	
	RF	95.92	92.60	93.95	88.59	98.27	94.35	0.096180
		(1.33)	(1.44)	(0.93)	(1.65)	(0.57)	(0.81)	
	AdaBoost	**96.18**	**93.70**	**94.71**	**89.92**	**98.43**	**95.00**	--
		(1.28)	(1.14)	(0.73)	(1.45)	(0.66)	(0.72)	
FDDT	SVM	96.10	93.88	94.76	90.03	98.61	95.05	0.11240
		(1.11)	(1.65)	(0.86)	(1.55)	(0.49)	(0.75)	
	NN	95.42	93.88	94.54	89.33	98.45	94.70	0.02837
		(0.79)	(1.56)	(0.92)	(1.68)	(0.53)	(0.82)	
	RF	95.96	92.13	93.65	88.18	98.30	94.16	0.001309
		(1.47)	(1.43)	(0.89)	(1.73)	(0.54)	(0.85)	
	AdaBoost	**97.03**	**94.06**	**95.23**	**91.15**	**98.80**	**95.61**	--
		(1.41)	(0.94)	(0.80)	(1.72)	(0.47)	(0.86)	
SCMM1	SVM	95.89	93.31	94.33	89.25	98.54	94.66	0.004556
		(1.69)	(1.35)	(0.93)	(1.87)	(0.58)	(0.94)	
	NN	95.53	93.74	94.48	89.30	98.50	94.68	0.002807
		(1.20)	(1.15)	(0.60)	(1.17)	(0.56)	(0.59)	
	RF	95.42	92.67	93.84	88.13	98.19	94.13	0.001147
		(1.20)	(1.12)	(0.81)	(1.58)	(0.37)	(0.79)	
	AdaBoost	**97.03**	**94.28**	**95.39**	**91.35**	**98.81**	**95.71**	--
		(1.09)	(0.99)	(0.77)	(1.52)	(0.53)	(0.76)	

The value inside the brackets () are the standard errors

The reported p-values have been computed between AdaBoost and other classifiers using F-measure criteria

Table 4.4 Performance comparison of the SVM, RF, NN and AdaBoost under three proposed encoding methods on predicting unbalanced Donor splice sites of HS3D datasets

Encoding Methods	Candidate classifier	UnBalanced Donor sites						p-value (AdaBoost)
		S_n	S_p	Q^9	Mcc	AUC	F-measure	
DTMM1	SVM	94.31	91.98	93.01	67.92	97.77	68.73	0.0001562
		(1.26)	(0.56)	(0.65)	(1.65)	(0.29)	(1.62)	
	NN	93.28	91.02	92.00	65.09	97.17	65.93	0.0001571
		(1.70)	(0.78)	(0.70)	(1.54)	(0.27)	(1.60)	
	RF	95.60	92.75	93.94	70.60	98.29	71.32	0.003598
		(1.31)	(0.71)	(0.65)	(1.99)	(0.24)	(2.00)	
	AdaBoost	**96.06**	**93.55**	**94.60**	**73.00**	**98.42**	**73.77**	--
		(1.27)	(0.55)	(0.54)	(1.62)	(0.26)	(1.63)	
FDDT	SVM	95.85	93.79	94.70	73.51	98.52	74.34	0.01556
		(0.72)	(0.55)	(0.42)	(1.59)	(0.26)	(1.63)	
	NN	95.96	93.61	94.63	73.10	98.42	73.89	0.001699
		(0.79)	(0.56)	(0.39)	(1.50)	(0.24)	(1.56)	
	RF	96.06	92.68	94.09	70.73	98.36	71.38	0.000157
		(0.91)	(0.54)	(0.40)	(1.35)	(0.22)	(1.41)	
	AdaBoost	**96.82**	**94.22**	**95.29**	**75.32**	**98.77**	**76.07**	--
		(0.99)	(0.38)	(0.33)	(1.10)	(0.21)	(1.11)	
SCMM1	SVM	96.03	93.98	94.89	74.15	98.58	74.98	0.001499
		(0.57)	(0.28)	(0.23)	(0.77)	(0.19)	(0.79)	
	NN	95.81	94.11	94.83	74.37	98.54	75.24	0.010170
		(1.31)	()0.41	(0.48)	(1.29)	(0.22)	(1.26)	
	RF	95.17	92.42	93.62	69.53	98.27	70.27	0.0001571
		(0.96)	(0.54)	(0.55)	(1.59)	(0.20)	(1.59)	
	AdaBoost	**96.71**	**94.54**	**95.45**	**76.20**	**98.86**	**77.00**	--
		(1.08)	(0.39)	(0.47)	(1.37)	(0.19)	(1.34)	

The value inside the brackets () are the standard errors
The reported p-values have been computed between AdaBoost and other classifiers using F-measure criteria

4.1.4 Performance Comparison with different state-of-the-art Methods

The performance results of other methods were shown in Table 4.5 and Table 4.6 for acceptor sites. In total, the selected proposed method, i.e. SCMM1-AdaBoost, significantly outperforms all the other methods in all the terms in both balanced and imbalanced acceptor datasets. However, it does not significantly outperform the FDDT-AdaBoost approach and does not perform better than the MSC+Pos-SVM approach in imbalanced data. But the difference is less than 1.5% which is a slight difference. Table

4.7 and Table 4.8 show the performance comparison of the proposed methods and other current methods for donor splice sites using balanced and imbalanced datasets. From Tables 4.7 and 4.8, it can be seen that SCMM1-AdaBoost significantly outperforms all the methods, while it does not statistically significantly outperform the FDDT-AdaBoost and MSC+Pos+APR-SVM methods and produce the same result approximately.

Table 4.5 Performance comparison of the proposed methods with others state of art methods on balanced Acceptor splice sites

Methods	**Balanced Acceptor sites**						**p-value**
	S_n	S_p	Q^9	Mcc	**AUC**	**F-measure**	*****
MM1-SVM	90.17	87.54	88.65	77.78	95.33	89.00	0.005062
	(2.01)	(2.15)	(1.06)	(2.06)	(0.42)	(1.02)	
Reduced MM1-SVM	90.83	88.02	89.27	78.91	95.48	94.36	0.005062
	(1.49)	(1.42)	(0.78)	(1.47)	(0.51)	(1.13)	
B-LSVM	90.90	88.16	89.34	79.13	95.88	89.67	0.005062
	(1.97)	(1.68)	(0.99)	(2.09)	(0.40)	(1.06)	
PN FDTF-SVM	91.18	88.47	89.64	79.71	96.12	89.96	0.005062
	(1.52)	(1.87)	(0.88)	(1.52)	(0.54)	(0.75)	
LVMM2	88.96	90.11	89.40	79.20	95.90	89.70	0.005062
	(1.70)	(1.08)	(1.24)	(1.48)	(0.43)	(0.85)	
MM2F-SVM	91.84	88.16	89.76	80.08	96.02	90.17	0.005062
	(1.77)	(1.00)	(0.66)	(1.57)	(0.49)	(0.82)	
MM2-RF	92.11	89.53	90.65	81.68	96.64	90.75	0.005062
	(1.27)	(0.90)	(0.62)	(1.33)	(0.48)	(0.68)	
MCM-SVM	92.46	90.80	91.56	83.28	97.21	91.70	0.005062
	(0.94)	(1.23)	(0.78)	(1.53)	(0.48)	(0.75)	
ECS-LSVM	91.74	90.00	90.78	81.76	96.64	90.95	0.005062
	(1.26)	(1.67)	(1.09)	(2.10)	(0.49)	(1.03)	
Sparse-SVM	92.71	89.86	91.10	82.62	96.94	91.41	0.005062
	(1.13)	(1.63)	(0.82)	(1.53)	(0.40)	(0.73)	
DM-SVM	92.64	90.73	91.54	83.41	97.31	91.76	0.005062
	(1.22)	(1.57)	(0.62)	(1.19)	(0.34)	(0.57)	
MSC+POS-SVM	94.69	92.78	93.61	87.49	98.30	93.80	0.7989
	(1.42)	(1.09)	(1.00)	(2.11)	(0.29)	(1.05)	
DTMM1-AdaBoost	93.72	91.04	92.16	84.82	97.41	92.48	0.00691
	(1.70)	(1.35)	(0.74)	(1.52)	(0.39)	(0.78)	
FDDT-AdaBoost	94.76	91.87	93.13	86.68	98.10	93.41	0.02842
	(0.69)	(0.94)	(0.65)	(1.33)	(0.37)	(0.66)	
SCMM1-AdaBoost*	**95.07**	**92.60**	**93.64**	**87.72**	**98.14**	**93.91**	--
	(1.16)	(1.40)	(0.77)	(1.40)	(0.33)	(0.69)	

Table 4.6 Performance comparison of the proposed methods with others state of art methods on unbalanced Acceptor splice sites

Methods	**UnBalanced Acceptor sites**						**p-value**
	S_n	S_p	Q^9	Mcc	**AUC**	**F-measure**	*
MM1-SVM	91.18	88.28	89.54	58.30	95.76	59.14	0.005062
	(2.56)	(0.54)	(1.15)	(1.99)	(0.57)	(1.66)	
Reduced MM1-SVM	90.72	88.23	89.35	58.01	95.66	58.90	0.005062
	(1.45)	(0.88)	(0.67)	(1.59)	(0.33)	(1.62)	
B-LSVM	91.81	88.20	89.82	58.58	96.04	59.30	0.005062
	(1.40)	(0.72)	(0.82)	(1.79)	(0.40)	(1.69)	
PN FDTF-SVM	92.15	88.35	90.03	59.06	96.14	59.74	0.005062
	(1.56)	(0.78)	(0.83)	(1.92)	(0.46)	(1.83)	
LVMM2	89.10	90.08	89.90	60.26	96.00	61.63	0.005062
	(1.15)	(0.43)	(0.32)	(0.87)	(0.43)	(0.84)	
MM2F-SVM	92.22	88.81	90.33	59.93	96.23	60.66	0.005062
	(1.43)	(0.55)	(0.72)	(1.52)	(0.55)	(1.41)	
MM2-RF	92.24	90.05	91.04	62.23	96.74	63.00	0.005062
	(1.62)	(0.69)	(0.81)	(1.83)	(0.50)	(1.74)	
MCM-SVM	92.33	90.92	91.48	64.28	97.26	65.25	0.005062
	(2.53)	(0.64)	(1.16)	(2.07)	(0.52)	(1.86)	
ECS-LSVM	91.15	89.47	90.22	60.55	96.41	61.53	0.005062
	(1.65)	(0.94)	(0.85)	(1.80)	(0.34)	(1.84)	
Sparse-SVM	92.64	89.47	90.89	61.48	96.81	62.22	0.005062
	(1.01)	(0.74)	(0.5)	(1.36)	(0.36)	(1.44)	
DM-SVM	92.81	90.44	91.51	63.56	97.16	64.39	0.005062
	(1.13)	(0.66)	(0.54)	(1.43)	(0.38)	(1.45)	
MSC+POS-SVM	94.93	92.76	93.68	70.21	98.24	71.03	0.009344
	(1.63)	(0.57)	(0.69)	(1.38)	(0.32)	(1.38)	
DTMM1-AdaBoost	93.96	91.04	92.35	65.56	97.42	66.30	0.005062
	(1.04)	(0.67)	(0.73)	(1.96)	(0.39)	(1.91)	
FDDT-AdaBoost	95.24	92.30	93.54	69.27	98.16	69.98	0.2845
	(1.40)	(0.56)	(0.57)	(1.42)	(0.40)	(1.42)	
SCMM1-AdaBoost*	**94.86**	**92.27**	**93.39**	**68.95**	**98.20**	**69.72**	--
	(1.25)	(0.61)	(0.55)	(1.54)	(0.27)	(1.56)	

Table 4.7 Performance comparison of the proposed methods with others state of art methods on balanced Donor splice sites

Methods	**Balanced Donor sites**						**p-value**
	S_n	S_p	Q^9	Mcc	**AUC**	**F-measure**	*
MM1-SVM	93.24	90.99	92.00	84.28	97.12	92.20	0.005062
	(1.88)	(1.77)	(1.01)	(2.07)	(0.57)	(1.04)	
Reduced MM1-SVM	93.92	91.24	92.36	85.22	97.32	92.67	0.005062
	(1.88)	(1.05)	(0.81)	(1.70)	(0.74)	(0.89)	
B-LSVM	94.28	90.77	92.30	85.11	97.33	92.65	0.005062
	(1.70)	(1.29)	(1.20)	(2.59)	(0.66)	(1.28)	
MM2F-SVM	93.70	91.60	92.50	85.34	97.33	92.72	0.005062
	(1.42)	(1.32)	(0.79)	(1.51)	(0.67)	(0.76)	
ECS-LSVM	95.67	90.95	92.86	86.73	97.71	93.30	0.005062
	(1.29)	(1.06)	(0.85)	(1.84)	(0.48)	(0.90)	
Sparse-SVM	94.64	91.95	93.07	86.64	97.73	93.38	0.005062
	(1.65)	(1.15)	(0.80)	(1.68)	(0.55)	(0.86)	
OLVWMM2	93.99	92.06	93.10	86.11	97.56	93.50	0.005062
	(1.70)	(1.64)	(1.26)	(2.58)	(0.62)	(1.27)	
P1-RF	95.64	92.20	93.56	87.92	97.81	94.02	0.005062
	(1.91)	(1.11)	(0.83)	(1.89)	(0.64)	(0.96)	
PN FDTF-SVM	95.28	92.67	93.74	87.99	97.94	94.05	0.005062
	(1.74)	(1.24)	(0.98)	(2.18)	(0.67)	(1.08)	
MM2-RF	95.92	91.32	93.17	87.39	97.97	93.71	0.005062
	(1.17)	(1.60)	(0.96)	(1.42)	(0.64)	(0.69)	
DM-SVM	95.17	92.13	93.36	87.34	98.05	93.73	0.005062
	(1.16)	(1.92)	(1.06)	(1.72)	(0.45)	(0.83)	
MCM-SVM	96.39	93.56	94.73	90.00	98.52	95.04	0.01246
	(1.13)	(0.94)	(0.69)	(1.45)	(0.57)	(0.72)	
MSC+POS+APR-SVM	96.92	94.67	95.62	91.63	98.84	95.85	0.06584
	(0.81)	(1.42)	(1.04)	(1.93)	(0.63)	(0.94)	
DTMM1-AdaBoost	96.18	93.70	94.71	89.92	98.43	95.00	0.01898
	(1.28)	(1.14)	(0.73)	(1.45)	(0.66)	(0.72)	
FDDT-AdaBoost	97.03	94.06	95.23	91.15	98.80	95.61	0.05934
	(1.41)	(0.94)	(0.80)	(1.72)	(0.47)	(0.86)	
SCMM1-AdaBoost*	97.03	94.28	95.39	91.35	98.81	95.71	--
	(1.09)	(0.99)	(0.77)	(1.52)	(0.53)	(0.76)	

Overall, the proposed methods exhibit good performance in both balanced and imbalanced datasets for the donor and acceptor site detection. Although the best proposed method has comparative prediction performance for imbalanced donor site in comparison with the high-accuracy MSC+Pos+APR-SVM method, it is less accurate in the imbalanced acceptor site.

Table 4.8 Performance comparison of the proposed methods with others state of art methods on unbalanced Donor splice sites

Methods	**UnBalanced Donor sites**						**p-value**
	S_n	S_p	Q^9	Mcc	**AUC**	**F-measure**	*
MM1-SVM	93.10	91.08	91.96	65.10	97.15	65.98	0.005062
	(1.89)	(0.48)	(0.90)	(1.66)	(0.35)	(1.50)	
Reduced MM1-SVM	93.96	91.08	92.34	65.63	97.32	66.37	0.005062
	(1.22)	(0.49)	(0.52)	(1.21)	(0.36)	(1.19)	
B-LSVM	93.99	90.96	92.23	65.43	97.34	66.15	0.005062
	(1.89)	(0.80)	(0.70)	(1.59)	(0.28)	(1.63)	
MM2F-SVM	93.21	91.39	92.20	65.85	97.44	66.75	0.005062
	(1.49)	(0.44)	(0.70)	(1.40)	(0.31)	(1.30)	
ECS-LSVM	94.92	90.86	92.56	65.78	97.69	66.33	0.005062
	(1.46)	(0.52)	(0.52)	(1.51)	(0.34)	(1.56)	
Sparse-SVM	94.42	91.61	92.84	67.11	97.80	67.86	0.005062
	(1.21)	(0.46)	(0.50)	(1.27)	(0.29)	(1.24)	
OLVWMM2	94.21	92.50	93.12	67.71	97.85	68.50	0.005062
	(1.33)	(0.69)	(0.66)	(1.05)	(0.32)	(1.40)	
P1-RF	95.35	92.25	93.56	69.25	97.80	69.94	0.005062
	(1.16)	(0.80)	(0.56)	(1.81)	(0.35)	(1.91)	
PN FDTF-SVM	95.21	92.78	93.86	70.43	98.07	71.22	0.005062
	(0.74)	(0.44)	(0.39)	(1.20)	(0.28)	(1.22)	
MM2-RF	95.56	91.65	93.28	67.55	97.95	68.12	0.005062
	(1.30)	(0.53)	(0.62)	(1.63)	(0.30)	(1.57)	
DM-SVM	94.60	92.34	93.30	68.71	98.00	69.51	0.005062
	(1.28)	(0.52)	(0.54)	(1.37)	(0.29)	(1.59)	
MCM-SVM	95.75	94.24	94.88	74.71	98.66	75.61	0.005062
	(1.45)	(0.45)	(0.68)	(1.42)	(0.25)	(1.39)	
MSC+POS+APR-SVM	96.78	94.89	95.69	77.28	98.93	78.11	0.01252
	(0.96)	(0.37)	(0.38)	(1.19)	(0.19)	(1.19)	
DTMM1-AdaBoost	96.06	93.55	94.60	73.00	98.42	73.77	0.00691
	(1.27)	(0.55)	(0.54)	(1.62)	(0.26)	(1.63)	
FDDT-AdaBoost	96.82	94.22	95.29	75.32	98.77	76.07	0.0166
	(0.99)	(0.38)	(0.33)	(1.10)	(0.21)	(1.11)	
SCMM1-AdaBoost*	96.71	94.54	95.45	76.20	98.86	77.00	--
	(1.08)	(0.39)	(0.47)	(1.37)	(0.19)	(1.34)	

The value inside the brackets () are the standard errors

The reported p-values have been computed between AdaBoost and other classifiers using F-measure criteria

4.1.5 Evaluation on the NN269 Dataset

We applied the proposed methods to the NN269 dataset for further evaluation. The performance metrics of AdaBoost, SVM, RF and NN with three encoding methods for both acceptor and donor sites were presented in Table 4.9 and Table 4.10. It is observed that the AdaBoost performs better than the other classifiers in acceptor sites, while it produces comparable results for donor sites in all the terms. The comparison result of the proposed methods with other state-of-the-arts methods on this dataset were given in Table 4.11. From Tables 4.11, it is seen that the proposed method for acceptor site outperforms all the current methods in terms of AUC and auPRC, while for donor site it outperforms all the other methods except the DPCH-SVM method in both of the terms.

Table 4.9 Performance comparison of SVM, RF, NN and AdaBoost under three proposed encoding methods on predicting acceptor splice sites of NN269 dataset

Encoding Methods	**Candidate classifier**	**Acceptor splice sites**						
		S_n	S_p	Q^9	Mcc	**AUC**	$auPRC$	**F-measure**
DTMM1	SVM	76.92	96.48	83.49	75.87	96.75	88.99	80.20
	NN	79.33	96.25	85.14	77.03	97.37	88.75	81.28
	RF	74.52	97.84	81.92	77.63	97.31	90.44	81.15
	AdaBoost	**83.65**	**97.28**	**88.28**	**82.48**	**98.56**	**93.91**	**85.71**
FDDT	SVM	79.81	96.14	85.46	77.10	97.52	90.87	81.37
	NN	78.85	96.82	84.87	78.05	97.61	90.77	82.00
	RF	71.63	98.75	79.92	78.15	97.88	91.56	80.98
	AdaBoost	**86.06**	**97.84**	**90.02**	**85.51**	**98.69**	**94.33**	**88.18**
SCMM1	SVM	78.37	97.05	84.56	78.27	98.10	92.21	82.12
	NN	79.33	97.50	85.28	80.08	98.24	92.62	83.54
	RF	72.60	98.30	80.59	77.53	97.58	91.21	80.75
	AdaBoost	**84.13**	**97.50**	**88.64**	**83.37**	**98.69**	**94.74**	**86.42**

Table 4.10 Performance comparison of SVM, RF, NN and AdaBoost under three proposed encoding methods on predicting donor splice sites of NN269 dataset

Encoding Methods	Candidate classifier	Donor splice sites						
		S_n	S_p	Q^9	Mcc	AUC	$auPRC$	F-measure
DTMM1	SVM	86.54	96.29	90.13	82.68	97.47	90.91	86.33
	NN	85.10	96.16	89.12	81.40	97.37	90.45	85.30
	RF	86.06	97.19	89.94	84.31	96.72	92.47	87.53
	AdaBoost	**87.98**	**96.68**	**91.18**	**84.51**	**97.03**	**92.34**	**87.77**
FDDT	SVM	87.02	96.16	90.43	82.75	98.04	91.73	86.40
	NN	89.42	96.29	92.07	84.69	98.00	90.90	87.94
	RF	87.98	96.80	91.21	84.78	97.83	91.48	87.98
	AdaBoost	**88.46**	**96.55**	**91.48**	**84.56**	**98.00**	**91.46**	**87.83**
SCMM1	SVM	88.46	96.68	91.51	84.84	98.21	93.12	88.04
	NN	87.98	96.55	91.16	84.23	98.18	92.42	87.56
	RF	87.02	98.08	90.72	87.02	97.95	92.00	89.60
	AdaBoost	**88.46**	**96.93**	**91.56**	**85.39**	**98.05**	**92.93**	**88.46**

Table 4.11 Comparison of different models on NN269 dataset

Methods	Acceptor splice sites		Donor splice sites	
	Performance Metrics		Performance Metrics	
	AUC	$auPRC$	AUC	$auPRC$
Reduced MM1-SVM	97.45	89.98	97.91	90.30
Weighted Degree Kernel	98.16	92.53	98.50	92.86
Weighted Degree Kernel with shift	98.65	94.36	98.13	92.47
DPCH-SVM	97.88	90.79	98.46	93.29
B-LSVM	97.79	91.24	97.95	92.00
MM2-RF	97.63	90.59	97.15	91.66
MCM-SVM	97.97	91.51	98.11	91.75
DTMM1-AdaBoost	98.56	93.91	97.03	92.34
FDDT-AdaBoost	98.69	94.33	98.00	91.46
SCMM1-AdaBoost	98.69	94.74	98.05	92.93

4.1.6 Online Prediction Server-HSSAda

Since the SCMM1-AdaBoost is observed to be superior over the other proposed approaches, we only included it in the server for prediction. The home page of the web server after execution of an example is exhibited in Figure 4.2. The prediction tool, HSSAda, is freely available at https://pashaei.shinyapps.io/hssada/.

The performance of the HSSAda was compared with several well-known existing tools using an independent test dataset, which is presented in Table 4.12 and Table 4.13. It can be seen that the proposed method outperforms others in all the terms.

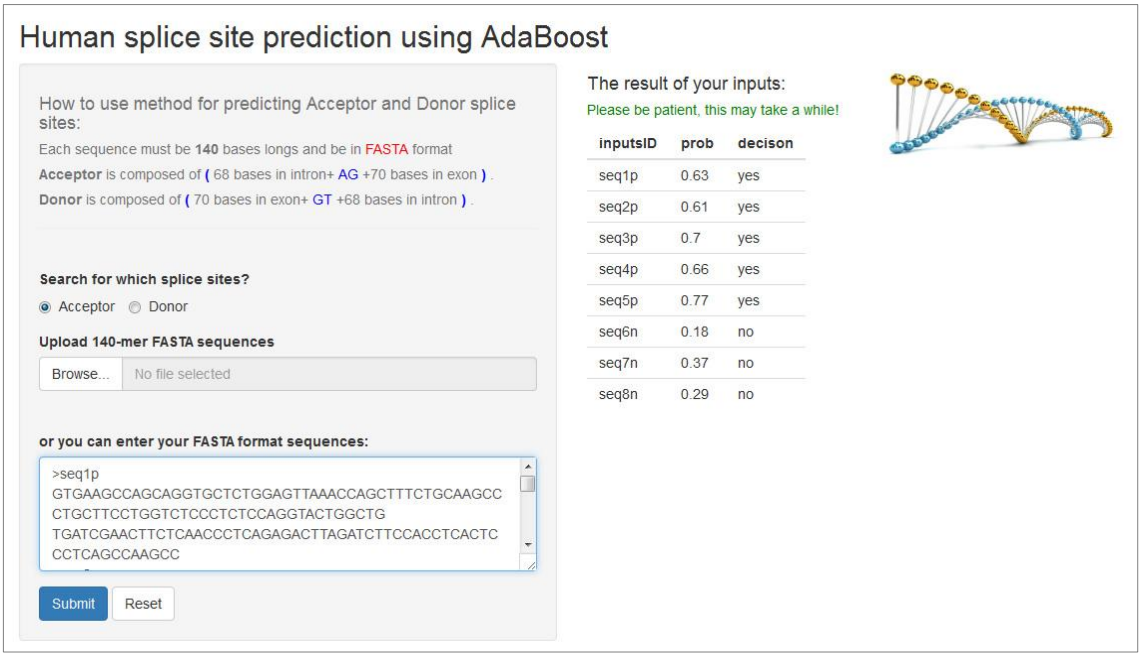

Figure 4.2 Snapshot of the developed web tool

Table 4.12 Performance comparison with other in-silico tools using the independent Acceptor test set

Methods	**Acceptor splice sites**				
	TPR	***TNR***	***Q^9***	***Mcc***	**F-measure**
WMM	100	83.12	88.07	70.45	74.77
MM1	100	84.38	88.95	72.06	76.19
MaxEntScan	100	83.75	88.51	71.24	75.47
NNSplice	85.00	91.88	87.94	72.52	78.16
Our proposed	100	91.88	94.25	83.27	86.02

Table 4.13 Performance comparison with other in-silico tools using the independent Donor test set

Methods	**Donor splice sites**				
	TPR	***TNR***	Q^9	***Mcc***	**F-measure**
WMM	95.00	77.51	83.71	59.30	65.52
MM1	97.50	80.62	86.19	65.52	70.91
MaxEntScan	95.00	85.00	88.82	69.19	74.51
NNSplice	85.00	89.38	87.00	68.26	74.73
Our proposed	100	93.12	95.14	85.46	87.91

4.2 Experimental Result using RF

4.2.1 Efficiency of RF as Feature Ranking Approach

The performance of selected attributes on balanced and unbalanced datasets have been shown in Figure 4.3. From the figure, it is possible to state that the accuracy of simple MM1-SVM has been improved by using feature ranking approaches. By considering balanced datasets (see Figure 4.3 (a) and (b)), it can be seen that both feature ranking methods have approximately the same accuracy on their optimal points. Additionally, the optimal points of both are equal in balanced acceptor and donor sites. The optimal point of balanced acceptor dataset and balanced donor dataset have been achieved by choosing 60% and 30% of top features using both of the feature ranking methods, respectively. Considering results for unbalanced datasets shown in the second row of the Fig. 4.3, the result of the RF ranking in acceptor sites (see Figure 4.3 (c)) is higher than the F-Score and the optimal point has been obtained using fewer numbers of attributes. In unbalanced donor splice sites (See Figure 4.3 (d)) F-Score shows better performance than the RF ranking method. So, on 4 datasets, the RF ranking method shows two equal, one win and one failure on its performance. As a result, on average it can be concluded that the RF feature ranking method is a good candidate for performing feature selection as preprocessing part on splice sites prediction methods.

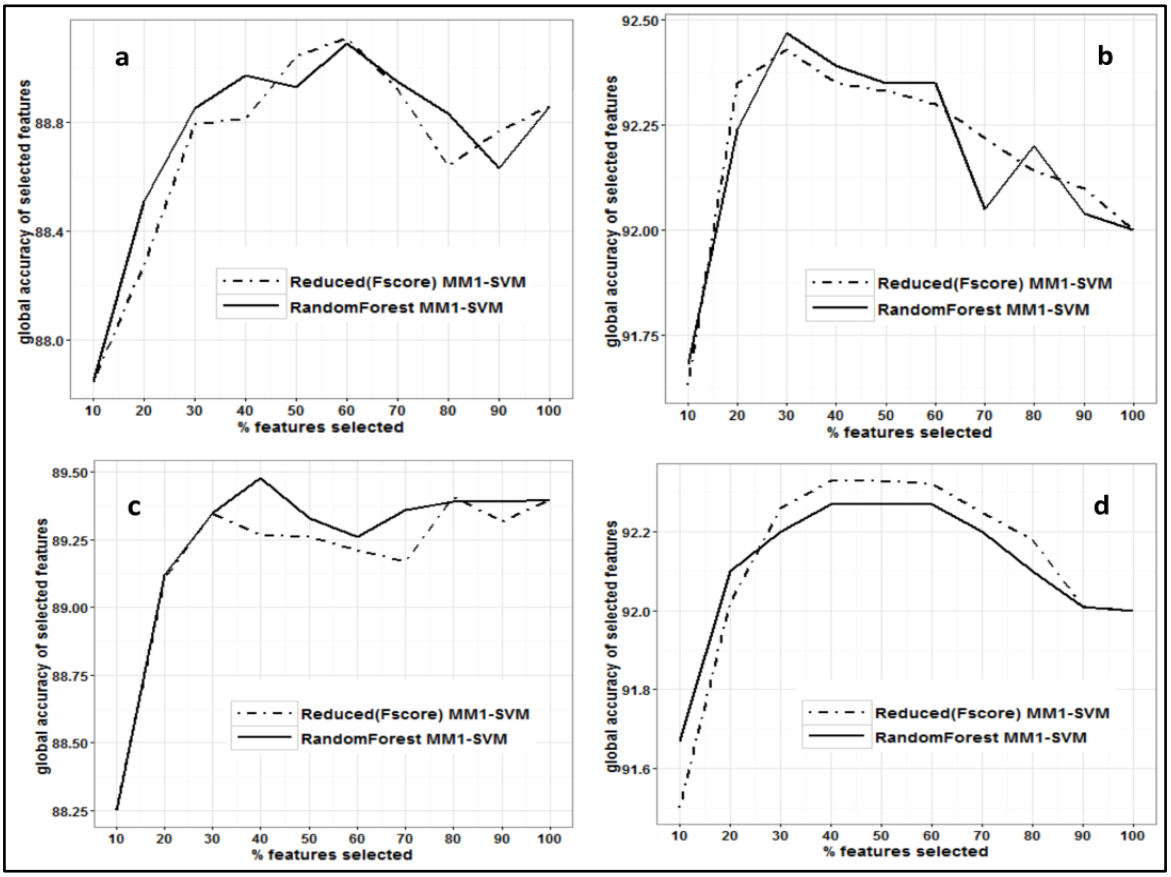

Figure 4.3 Global accuracy of different percentage of selected features using F-score feature ranking and random forest feature ranking methods on a) Balanced acceptor splice sites, b) Balanced donor splice sites, c) Unbalanced acceptor splice sites and d) Unbalance donor splice sites datasets for assessing performance of MM1-SVM method

4.2.2 Efficiency of RF as Classifier

The performance results of classification have been shown in Table 4.14. Since different training data are obtained due to employing different encoding methods, we considered each row of the table as an independent dataset. Therefore, our experiment utilized 18 different datasets which 9 of them belong to the acceptor sites while the remains belong to the donor sites. The performance was estimated using various measures. However, we preferred $F-measure$ to make statistical comparisons (reported P-value) between SVM and RF. We should take into account that we could not carry out the statistical evaluation on NN269 dataset due to default separation between the training set and test set. However, we consider their result significant when the difference in F-measure became more than 1.50% between SVM and RF.

Table 4.14 Comparison of classification performance of SVMs and RFs using Markovian encoding methods

Dataset	Classifier	Encoding Methods	Support Vector Machine (SVM)						
			S_n	S_p	Q^9	Mcc	AUC	F-measure	CPU time
Balanced Acceptor Sites (HS3D)	SVM	MM1	90.22	87.79	88.86	78.06	95.43	89.14*	1556.25
		MM2	91.42	88.17	89.60	79.65	96.00	89.95*	1274.84
		MCM	92.87	88.35	90.28	81.32	96.52	90.82	1301.68
	RF	MM1	91.69	89.10	90.24	80.84	96.36	90.52*	484.89
		MM2	92.16	89.47	90.65	81.68	96.62	90.94*	500.98
		MCM	91.92	89.67	90.66	81.63	96.62	90.90	557.02
Balanced Donor sites (HS3D)	SVM	MM1	93.27	91.12	92.01	84.43	97.15	92.27	1064.89
		MM2	93.37	91.59	92.36	85.00	97.34	92.55*	793.94
		MCM	95.67	93.16	94.20	88.94	98.18	94.52	1006.01
	RF	MM1	95.40	91.00	92.78	86.50	97.72	93.35	472.25
		MM2	95.77	91.29	93.09	87.17	97.97	93.67*	467.84
		MCM	95.44	92.00	93.41	87.52	97.93	93.84	622.96
UnBalanced Acceptor sites (HS3D)	SVM	MM1	90.77	88.27	89.39	58.04	95.78	58.95*	1678.18
		MM2	91.90	88.97	90.29	60.04	96.25	60.83*	1332.50
		MCM	92.88	89.00	90.69	60.70	96.66	61.34*	1388.43
	RF	MM1	91.87	89.52	90.59	61.09	96.52	61.96*	528.20
		MM2	92.17	89.99	90.97	62.21	96.73	63.09*	525.89
		MCM	91.94	89.69	90.71	61.48	96.64	62.56*	582.91
UnBalanced Donor sites (HS3D)	SVM	MM1	93.24	91.00	91.99	65.02	97.18	65.54	1146.47
		MM2	93.41	91.37	92.27	65.94	97.38	66.80	837.84
		MCM	95.36	93.14	94.09	71.49	98.22	72.30*	1059.76
	RF	MM1	95.06	90.72	92.52	65.55	97.75	66.07	472.68
		MM2	95.70	91.55	93.25	67.79	97.96	68.33	454.23
		MCM	94.93	92.12	93.32	68.65	98.02	69.38*	620.38
UnBalanced Acceptor sites (NN269)	SVM	MM1	75.96	96.71	82.84	75.74	97.52	80.00	336.60
		MM2	79.33	97.05	85.23	78.94	97.92	82.71	301.08
		MCM	80.77	97.73	86.31	81.65	98.43	84.85	297.36
	RF	MM1	70.19	97.50	78.85	73.67	97.33	77.66	28.91
		MM2	71.63	97.96	79.89	75.92	98.01	79.47	28.08
		MCM	69.71	98.30	78.55	75.51	97.69	78.80	34.82
UnBalanced Donor Sites (NN269)	SVM	MM1	83.65	96.16	88.13	80.39	97.82	84.47	54.95
		MM2	83.17	96.16	87.80	80.05	97.33	84.18	52.21
		MCM	90.87	95.65	92.85	84.37	98.06	87.70	58.24
	RF	MM1	85.58	96.55	89.51	82.56	97.86	86.20	4.37
		MM2	85.10	96.68	89.20	82.51	97.84	86.13	4.20
		MCM	88.01	96.42	91.21	83.96	97.70	87.35	5.18

The t-test have been computed between SVM and RF classifiers using F-measure criteria. The * show that the performance of RF is significantly betther than SVM.

According to the result, in total RF outperformed SVM statistical significantly (at the 0.01 a value) in 8 datasets, RF nominally (that is, not essentially statistically significant) outperforms SVM in 4 datasets, SVM nominally outperforms RF in 2 datasets, and in 4 datasets SVM outperforms RF statistically significant. So, considering 18 datasets, RF performs better than SVM in 12 datasets. In terms of computational efficiency, as can be seen from CPU time column, RFs performed much faster than SVM due to parameter optimization process that was composed to SVM.

In addition, the classification results of proposed methods MM1-RF, MM2-RF and MCM-RF compared with these of MM1-SVM [7], Reduced MM1-SVM [1], SVM-B [5], LVMM2 [11], MM2F-SVM [13] and MCM-SVM [14] methods using Q^9 criteria for HS3D dataset and $auPRC$ for NN269 dataset in Figure 4.4. The result of the LVMM2 was taken from [11].

From Figure 4.4, considering both balanced datasets, the proposed method MM1-RF outperformed MM1-SVM, Reduced MM1-SVM, SVM-B and MM2F-SVM for both acceptor (Figure 4.4 (a)) and donor splice site (Figure 4.4 (b)), but could not show better performance than MCM-SVM. Two other proposed methods, MM2-RF and MCM-RF performed better than MM1-RF for both acceptor and donor sites. In balanced acceptor splice site (Figure 4.4 (a)), MM2-RF and MCM-RF showed the same performance and both of them could outperform other methods. In balanced donor site (Figure 4.4 (b)), MCM-RF performed better than MM2-RF and MM1-RF and could outperform all of the other methods except MCM-SVM. Considering unbalanced acceptor dataset (Figure 4.4 (c)), we can see that MM1-RF outperformed the MM1-SVM, Reduced MM1-SVM and SVM-B and produce a comparable result with LVMM and MM2F-SVM. The MCM-RF method performed better than MM1-RF and could outperform LVMM and MM2F-SVM. The MM2-RF method performed better than MCM-RF and outperformed all methods significantly and stood out as the best method on unbalanced acceptor splice sites. In the unbalance donor site (Figure 4.4 (d)), the MM1-RF outperformed MM1-SVM, Reduced MM1-SVM, SVM-B, and MM2F-SVM. The MM2-RF performed better than MM1-RF and could produce comparable results with LVMM. The MCM-RF performed slightly better than the MM2-RF and could outperform all the methods except the MCM-SVM same as the MM2-RF. In comparison to LVMM2, the proposed methods MM2-RF and MCM-RF performed slightly better than LVMM2.

However, determining the associated threshold parameters of the LVMM [11] are difficult [12]. The proposed method has less complexity in comparison to LVMM2.

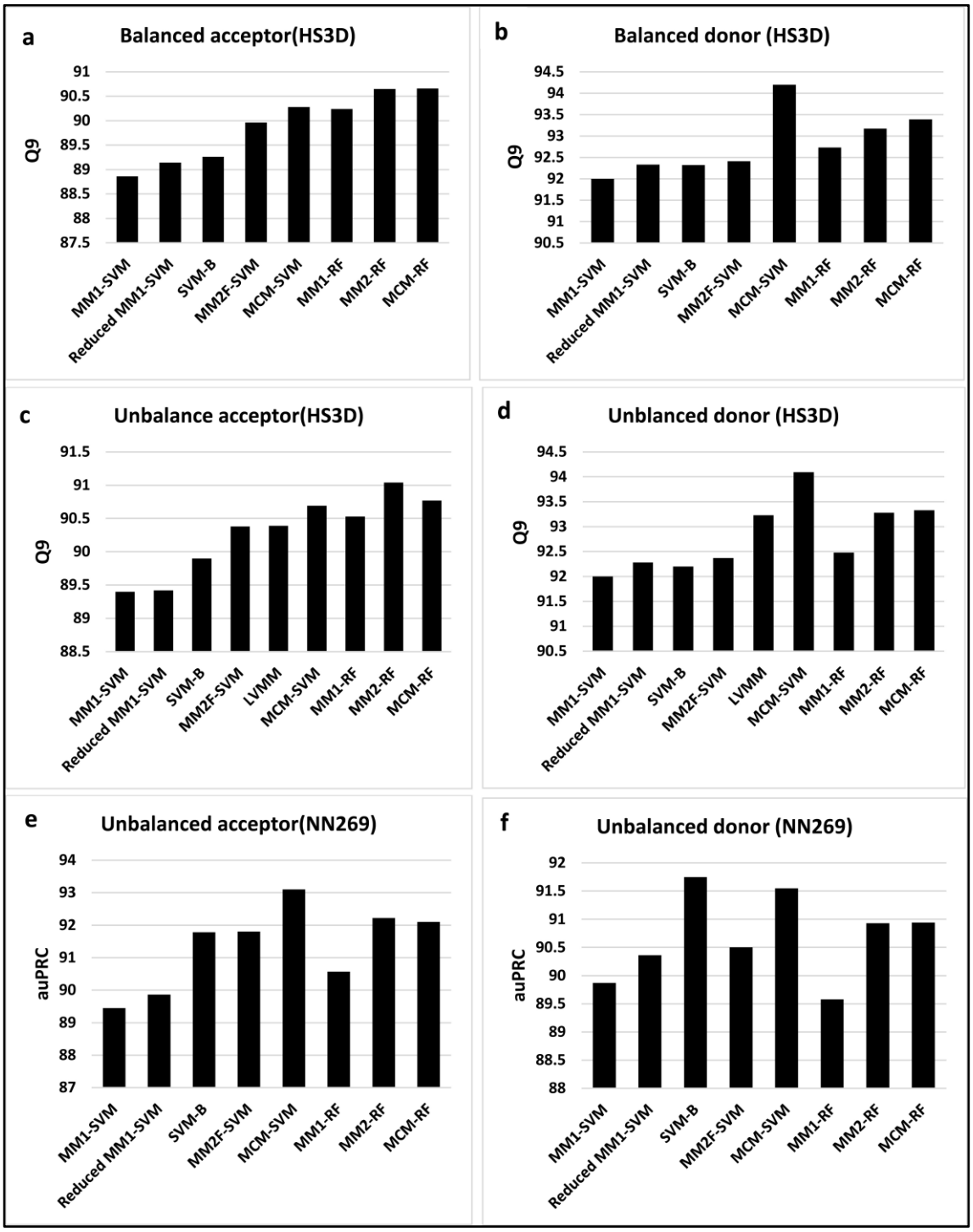

Figure 4.4 Classification performance of the different state-of-the-art methods for both HS3D and NN269 datasets

The overall performance comparison of the proposed methods can be summarized in this way. Considering the balanced acceptor dataset, MM2-RF and MCM-RF showed the best performance. The MCM-SVM method illustrated better accuracy than the proposed methods on balanced donor splice sites. Considering unbalanced datasets, the MM2-RF outperformed all the methods on acceptor site and again MCM-SVM showed higher accuracy in unbalanced donor sites. We can state that our proposed methods are definitely more suitable for acceptor sites than donor sites. Additionally, considering the performance of RF along with SVM using the same encoding methods, the proposed methods in most of the cases performed better.

In order to estimate the consistency of the proposed methods, we performed an additional evaluation on the NN269 dataset. For acceptor sites (Figure 4.4 (e)), auPRC of the MM1-RF is better than MM1-SVM and Reduced MM1-SVM. Besides, the MM2-RF performed better than MM2F-SVM and SVM-B. The MCM-RF outperformed all of the methods but MCM-SVM performed better than the proposed methods. For the donor sites (Figure 4.4 (f)), the auPRC of the MM1-RF method is lower than other available models. The MM2-RF and MCM-RF showed the same accuracy in term of auPRC. Both of them outperformed all methods except SVM-B and MCM-SVM methods. Overall, the proposed methods produced good results for the NN269 dataset.

4.3 Experimental Result using SVM

The results of balanced and unbalanced datasets on both acceptor and donor splice sites are given in the Table 4.15, Table 4.16, Table 4.17, and Table 4.18, respectively. The DMM2-SVM outperforms all the methods clearly for both acceptor and donor splice site on the balanced dataset, except the MCM-SVM method in the unbalanced dataset. For acceptor and donor sites, the Q9 score, Mcc and AUC of DMM2-SVM is clearly better than those of other methods. Also, DMM2-SVM performs better than MM1-SVM, Reduced MM1-SVM, SVM-B, MM2F-SVM, PN-FDTF-SVM, LVMM2, ESC-LSVM, Sparse-SVM, and DM-SVM in terms of sensitivity and specificity. In comparison with MCM-SVM, our method performs better in all the terms for acceptor and donor site in balance dataset sites and produces approximately the same results for both sites in unbalanced dataset.

Table 4.15 Performance comparison of the DMM2-SVM with others state of art methods on balanced Acceptor splice sites

Methods	**Balanced Acceptor splice sites**				
	S_n	S_p	Q^9	Mcc	**AUC**
MM1-SVM	90.17	87.54	88.65	77.78	95.33
Reduced MM1-SVM	90.83	88.02	89.27	78.91	95.48
B-LSVM	90.90	88.16	89.34	79.13	95.88
PN FDTF-SVM	91.18	88.47	89.64	79.71	96.12
LVMM2	88.96	90.11	89.40	79.20	95.90
MM2F-SVM	91.84	88.16	89.76	80.08	96.02
DM-SVM	92.64	90.73	91.54	83.41	97.31
MCM-SVM	92.46	90.80	91.56	83.28	97.21
DMM2-SVM	**94.16**	**91.91**	**92.82**	**86.13**	**97.86**

Table 4.16 Performance comparison of the proposed methods DMM2-SVM with others state of art methods on unbalanced Acceptor splice sites

Methods	**UnBalanced Acceptor splice sites**				
	S_n	S_p	Q^9	Mcc	**AUC**
MM1-SVM	91.18	88.28	89.54	58.30	95.76
Reduced MM1-SVM	90.72	88.23	89.35	58.01	95.66
B-LSVM	91.81	88.20	89.82	58.58	96.04
PN FDTF-SVM	92.15	88.35	90.03	59.06	96.14
LVMM2	89.10	90.08	89.90	60.26	96.00
MM2F-SVM	92.22	88.81	90.33	59.93	96.23
DM-SVM	92.81	90.44	91.51	63.56	97.16
MCM-SVM	92.33	90.92	91.48	64.28	97.26
DMM2-SVM	**94.06**	**90.37**	**91.93**	**64.18**	**97.46**

Table 4.17 Performance comparison of the proposed methods DMM2-SVM with others state of art methods on balanced Donor splice sites

Methods	**Balanced Donor splice sites**				
	S_n	S_p	Q^9	Mcc	**AUC**
MM1-SVM	93.24	90.99	92.00	84.28	97.12
Reduced MM1-SVM	93.92	91.24	92.36	85.22	97.32
B-LSVM	94.28	90.77	92.30	85.11	97.33
MM2F-SVM	93.70	91.60	92.50	85.34	97.35
OLVWMM2	93.99	92.06	93.10	86.11	97.56
DM-SVM	95.17	92.13	93.36	87.34	98.05
PN FDTF-SVM	95.28	92.67	93.74	87.99	97.94
MCM-SVM	96.39	93.56	94.73	90.00	98.52
DMM2-SVM	**95.96**	**94.06**	**94.85**	**90.05**	**98.54**

Our method has less parameter to tune in comparison with the LVMM2 (OLVWMM2) and the MCM-SVM approaches that have to be specified before using and the results of our proposed method are better too. It can be concluded that the DMM2-SVM is an efficient approach in identification of Human splice site on both balanced and unbalanced datasets. Besides, our proposed encoding method does not have any parameters to tune. It makes our method easier to use than others.

Table 4.18 Performance comparison of the proposed methods DMM2-SVM with others state of art methods on unbalanced Donor splice sites

Methods	**Balanced Donor splice sites**				
	S_n	S_p	Q^9	Mcc	**AUC**
MM1-SVM	93.10	91.08	91.96	65.10	97.15
Reduced MM1-SVM	93.96	91.08	92.34	65.63	97.32
B-LSVM	93.99	90.96	92.23	65.43	97.34
MM2F-SVM	93.21	91.39	92.20	65.85	97.44
OLVWMM2	94.21	92.50	93.12	67.71	97.85
DM-SVM	94.60	92.34	93.30	68.71	98.00
PN FDTF-SVM	95.21	92.78	93.86	70.43	98.07
MCM-SVM	95.75	94.24	94.88	74.71	98.66
DMM2-SVM	**95.14**	**93.94**	**94.47**	**73.48**	**98.48**

CHAPTER 5

DISCUSSION AND CONCLUSION

It can be seen from the literature that SVM-based classification techniques are frequently used because of their outstanding performance [40]. The accuracy of the SVM largely depends on choosing kernels and its parameters. Because the parameter tuning of the SVM can be a laborious task when multiple parameters are involved in it, it is arguable whether the SVM is a suitable method for genome-wide splice sites prediction [11]. Besides, efficient DNA encoding methods are essential to convert the DNA sequences to feature vectors in order to make them utilizable for the SVM. Each encoding method has its own advantage and disadvantages. For instance, encoding approaches of the LVMM2 and the MCM-SVM methods contain several thresholds that make these methods difficult to use in spite of their good performance. The MM2 encoding approach in the MM2F-SVM method is computationally expensive [12], while the idea of a hybrid of triplet nucleotide with MM1 as encoding approach in the DM-SVM method only performs slightly better than the LVMM2 method in donor splice sites detection. The MSC+Pos(+APR)-SVM method is the most outstanding approach in Human splice site prediction domain with regard to its high-throughput accuracy. However, the number of useless and inhibiting features that is produced just by the MSC encoding approach is too many. Additionally, because of the existence of several parameters that should be tuned in feature extraction section, the computation cost is high. Hence, improving the performance of the approaches for high prediction accuracy and capability of them for applying in the whole genome are still need.

In this thesis, we proposed novel DNA encoding methods by an efficient combination of several former proven successful encoding methods, in order to provide more informative features. The main advantages of the proposed methods to current state-of-

the-art methods are that they have no parameters to tune and they are easy to use. Besides, the proposed methods exhibit good prediction accuracy for both donor and acceptor splice sites.

In the first part of this thesis, we introduced three novel DNA encoding methods, which have been obtained by an efficient combination of previously existing DNA encoding methods in order to provide more information from DNA sequences. Besides, the AdaBoost classifier has been proposed in splice site prediction domain. It exhibited higher accuracy than the state-of-art SVM classifier in most of the cases when it was combined with the proposed DNA encoding methods. Our methods showed better performance in comparison with eleven currently available methods in the literature on HS3D dataset. Due to high prediction accuracy and existence of minimum tuning parameters, it can be concluded that our approaches are simple, efficient and easy to use. We also utilized the benchmark NN269 dataset to examine the reproducibility and stability of proposed approaches. Results showed that our methods are efficient in splice site prediction. In addition, the proposed methods are believed to contribute to the detection of unknown splice sites of whole genome and be extended for predicting other specific splice sites such as alternative splice sites in the DNA sequences. The relevant web server will help the biological community for easy detection of splice sites.

In the second part, we study RF as a new classifier and feature selection method in Human splice site prediction domain. Since a large number of features are used to describe structures or processes in biology, the elimination of irrelevant and redundant information provide useful biological knowledge for human experts. F-score feature ranking method is a simple and efficient method that is used in splice site prediction domain frequently. We have investigated the efficiency of RF feature ranking method by comparing it with F-score to show the capability of RF as a feature selection in Human splice sites identification. The results show that RF feature ranking is a useful method in human splice sites prediction.

SVM has been most commonly used in prediction of splice sites due to its high performance. But existing of the parameters that have to be set before using it, such as penalty parameter, the kernel type, and kernel parameters make it time-consuming process, causing to question whether SVM is a suitable method for genome-wide splice sites prediction [11]. We employ RF as another extremely successful classifier in this thesis. One of the main advantages of RF-based methods in comparison to SVM-based

methods is that it does not need tuning step in contrary to SVM and it is really fast with high performance. By combining RF with three up-to-date encoding methods (MM1, MM2, and MCM), we show that the proposed methods perform approximately the same and often better than the SVM-based methods. In addition, the proposed methods are simple, fast, easy to use and can be applied to large-scale Human Genome data for identifying splice sites. As a future study, these methods can also be utilized for identification of other regulatory regions such as translation initiation sites and promoters.

REFERENCES

[1] Baten, A. Halgamuge, S. and Chang, B., (2008). "Fast splice site detection using information content and feature reduction", BMC Bioinformatics, 9(Suppl 12).

[2] Li, J. Wang, L. Wang, H. Bai, L. and Yuan, Z., (2012). "High-accuracy splice sites prediction based on sequence component and position features", Genetics and Molecular Research, 11: 3432-3451.

[3] Sonnenburg, S. Schweikert, G. Philips, P. Behr, J. and Ratsch, G., (2007). "Accurate splice site prediction using support vector machines", BMC Bioinformatics, 88(Suppl 10).

[4] Tazi, J. Bakkour, N. and Stamm, S., (2009). "Alternative splicing and disease", Biochimica et Biophysica Acta, 1792: 14-26.

[5] Zhang, Y. Chu, C.-H. Chen, Y. Zha, H. and Ji, X., (2006). "Splice site prediction using support vector machines with a Bayes kernel", Expert Systems with Applications, 30: 73-81.

[6] Burge, C. and Karlin, S., (1997). "Predictions of complete gene structures in human genomic DNA", Journal of Molecular Biology, 9: 499-509.

[7] Baten, A. Chang, B. Halgamuge, S. and Li, J., (2006). "Splice site identification using probabilistic parameters and SVM classification", BMC Bioinformatics, 7(Suppl 5).

[8] Reese, M. Eeckman, F. Kupl, D. and Haussler, D., (1997). "Improved splice site detection in Genie", Journal of Computational Biology, 4: 311-324.

[9] Hebsgaard, S.M. Korning, P.G. Tolstrup, N. Engelbrecht, J. Rouzé, P. and Brunak, S., (1996). "Splice site prediction in Arabidopsis Thaliana pre-mRNA by combining local and global sequence information", nucleic acids research, 24: 3439-3452.

[10] Loi, H.S. and Rajapakse, J.C., (2003). "Splice Site Detection with a Higher-Order Markov Model Implemented on a Neural Network", Genome Informatics, 14: 64-72.

[11] Zhang, Q. Peng, Q. Zhang, Q. Yan, Y. Li, K. and Li, J., (2010). "Splice site prediction of human genome using Length-variable Markov model and feature selection", Expert Systems with Applications, 37: 2771-2782.

[12] Wei, D. Zhang, H. Wei, Y. and Jiang, Q., (2013). "A Novel Splice Site Prediction Method using Support Vector Machine", Journal of Computational Information Systems, 9: 8053-8060.

[13] Maji, S. and Garg, D., (2014). "Hybrid Approach Using SVM and MM2 in Splice Site Junction Identification", Current Bioinformatics, 9: 76-85.

[14] Goel, N. Singh, S. and Aseri, T.C., (2015). "An improved method for splice site prediction in DNA sequences using support vector machines", Procedia Computer Science, 57: 358-367.

[15] Bari, A.T.M.G. Reaz, M.R. Choi, H.J. and Jeong, B.S., (2012). "Survey on Nucleotide Encoding Techniques and SVM Kernel Design for Human Splice Site Prediction", Interdisciplinary Bio Central, 4: 1-6.

[16] Nassa, T. and Singh, S., (2013). "Neural Network Based Systems for Splice Site Detection: A Review", International Journal of Advanced Research in Computer Science and Software Engineering, 3: 604-608.

[17] Noordewier, M.O. Towell, G.G. and Shavlik, J.W., (1991). "Training Knowledge-Based Neural Networks to Recognize Genes in DNA Sequences", Advances in Neural Information Processing Systems, 3.

[18] Michie, D. Spiegelhalter, D.J. and Taylor, C.C., (1994). Machine Learning, Neural and Statistical Classification: Ellis Horwood.

[19] Rampone, S., (1998). "Recognition of splice junctions on DNA sequences by BRAIN learning algorithm", Bioinformatics, 14: 676-684.

[20] Yin, M. and Wang, J., (1998). "Algorithms for splicing junction donor recognition in genomic DNA sequences", IEEE Internat. Joint Symp. Intell. Syst: 169–176.

[21] Thivierge, J.P. and Shultz, T.R., (2002). "Finding relevant knowledge: KBCC applied to DNA splice-junction determination",IEEE International Joint Conference on Neural Networks: 1401-1405.

[22] Fahlman, S.E. and Lebiere, C., (1989). "The Cascade-Correlation Learning Architecture", Advances in Neural Information Processing Systems, 2: 525-532.

[23] Lorena, A. and Carvalho, A.d., (2003). "Human Splice Site Identifications with Multiclass Support Vector Machines and Bagging", Artificial Neural Neural Networks and Neural Information Processing – ICANN/ICONIP, 2714.

[24] Lorena, A.C. and Carvalho, A.C.P.L.F., (2004). "Evaluation of noise reduction techniques in the splice junction recognition problem", Genetics and Molecular biology, 27: 665-672.

[25] Lumini, A. and Nanni, L., (2006). "Identifying splice-junction sequences by hierarchical multiclassifier",Pattern Recognition Letters: 1390-1396.

[26] Nantasenamat, C. Naenna, T. Ayudhya, C.I.N. and Prachayasittikul, V., (2005). "Recognition of DNA splice junction via machine learning approaches", EXCLI Journal, 4: 114-129.

[27] Damasevicius, R., (2008). "Splice site recognition in DNA sequences using k-mer frequency based mapping for support vector machine with power series kernel",International Conference on Complex Intelligent and Software Intensive Systems: 687-692.

[28] Kerdprasop, N. and Kerdprasop, K., (2010). "A High Recall DNA Splice Site Prediction Based on Association Analysis",10 th WSEAS international conference on Applied computer science(ACS10): 484-489.

[29] Salekdeh, A.Y. and Wiese, K.C., (2011). "Improving splice-junctions classification employing a novel encoding schema and decision-tree",2011 IEEE Congress of Evolutionary Computation (CEC) New Orleans, USA: IEEE 1302 - 1307.

[30] Htike, Z.Z. and Win, S.L., (2013). "Classification of eukaryotic splice-junction genetic sequences using averaged one-dependence estimators with subsumption resolution",4th International Conference on Computational Systems-Biology and Bioinformatics(CSBio2013): Elsevier: 36-43.

[31] Maji, P. and Paul, S., (2014). Neural Network Tree for Identification of Splice Junction and Protein Coding Region in DNA, ed. Scalable Pattern Recognition Algorithms. Switzerland: Springer 45-66.

[32] Blake, C. and Merz, C., (1998). The UCI Repository of Machine Learning Databases, ed^eds. Irvine, CA: University of California, Department of Information and Computer Science.

[33] Staden, R., (1984). "Computer methods to locate signals in nucleic acid sequences", nucleic acids research, 12: 505-519.

[34] Tavares, L.G. Lopes, H.S. and Lima, C.R.E., (2009). "Evaluation of weight matrix models in the splice site junction recognition problem", Bioinformatics and Biomedicine Workshop, 1: 14-19.

[35] Zhang, M. and Marr, T., (1993). "A weight array method for splicing signal analysis", computational biosciences, 9: 499-509.

[36] Chen, Y.W. and Lin, C.J., (2006). Combining SVMs with Various Feature Selection Strategies, I. Guyon Gunn, S. Nikrevesh, M. ve Zadeh, L., ed. Feature Extraction Studies in Fuzziness and Soft Computing, . New York: Springer, 315-324.

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

[38] Meher, P.K. Sahu, T.K. and Rao, A.R., (2016). "Prediction of donor splice sites using random forest with a new sequence encoding approach", BioData Mining, 9.

[39] Pashaei, E. Ozen, M. and Aydin, N., (2016). "Splice site identification in human genome using random forest", Health and Technology: 1-12.

[40] Bari, A.T.M.G. Reaz, M.R. and Jeong, B.-S., (2014). "Effective DNA Encoding for Splice Site Prediction Using SVM", Communications in Mathematical and in Comuter Chemistry (MATCH), 71: 241-258.

[41] Wei, D. Zhuang, W. Jiang, Q. and Wei, Y., (2012). A new classification method for human gene splice site prediction, J. He Liu, X. Krupinski, E. ve Xu, G., ed. Health Information Science Springer Berlin Heidelberg, 121-130.

[42] Jian, X. Boerwinkle, E. and Liu, X., (2014). "In silico tools for splicing defect prediction: a survey from the viewpoint of end users", GENETICS in MEDICINE, 16: 497-503.

[43] Shapiro, M. and Senapathy, P., (1987). "RNA splice junctions of different classes of eukaryotes: sequence statistics and functional implications in gene expression", nucleic acids research, 15: 7155–7174.

[44] Rogozin, I.B. and Milanesi, L., (1997). "Analysis of donor splice signals in different Eukaryotic organisms", J. Mol. Evol, 45: 50-59.

[45] Brunak, S. Engelbrecht, J. and Knudsen, S., (1991). "Prediction of human mRNA donor and acceptor sites from the DNA sequence", Journal of Mol Biol, 220.

[46] Tolstrup, N. Rouzé, P. and Brunak, S., (1997). "A branch point consensus from Arabidopsis found by non-circular analysis allows for better prediction of acceptor sites", nucleic acids research, 25: 3159-3163.

[47] Solovyev, V.V. Salamov, A.A. and Lawrence, C.B., (1994). "Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames", nucleic acids research, 22: 5156-5163.

[48] Kleffe, J. Hermann, K. Vahrson, W. Wittig, B. and Brendel, V., (1996). "Logitlinear models for the prediction of splice sites in plant pre-mRNA sequences", nucleic acids research, 24: 4709-4718.

[49] Pertea, M. Lin, X. and Salzberg, S., (2001). "GeneSplicer: a new computational method for splice site prediction", nucleic acids research, 29: 1185-1190.

[50] Yeo, G. and Burge, C., (2004). "Maximum entropy modeling of short sequence motifs with application to RNA splicing signals", J Comput Biol, 11: 377-394.

[51] Dogan, R.I. Getoor, L. Wilbur, W.J. and Mount, S.M., (2007). "SplicePort—An interactive splice-site analysis tool", nucleic acids research, 35: 285-291.

[52] Desmet, F.O. Hamroun, D. Lalande, M. Collod-Beroud, G. Claustres, M. and Beroud, C., (2009). "Human Splice Finder: an online bioinformatics tool to predict splicing signals", nucleic acids research, 37.

[53] Yin, M. and Wang, J., (2001). "Effective hidden Markov models for detecting splicing junction sites in DNA sequences", Information Sciences, 139: 139-163.

[54] Cai, D. Delcher, A. Kao, B. and Ksif, S., (2000). "Modeling splice sites with Bayes networks", Bioinformatics, 16: 152-158.

[55] Chen, T.M. Lu, C.C. and Li, W.H., (2005). "Prediction of splice sites with dependency graphs and their expanded Bayesian networks", Bioinformatics, 21: 471-482.

[56] Rajapakse, J. and Ho, L., (2005). "Markov encoding for detecting signals in genomic sequences", IEEE-Acm Transactions on Computational Biology and Bioinformations, 2: 131-142.

[57] Marashi, S. Goodarzi, H. Sadeghi, M. Eslahchi, C. and Pezeshk, H., (2006). "Importance of RNA secondary structure information for yeast donor and acceptor splice site prediction by neural networks", Computational Biology and Chemistry, 30: 50-57.

[58] Lopes, H.S. Lima, C.R.E. and Murata, N.J., (2007). "A configware approach for high-speed parallel analysis of genomic data", Journal of Circuits Systems and Computers, 16: 527-540.

[59] Salekdeh, A.Y. and Wiese, K.C., (2011). "Improving Splice-Junctions Classification employing a Novel Encoding Schema and Decision-Tree", Evolutionary Computation (CEC): 1302 - 1307.

[60] Bin, W. and Jing, Z., (2014). "A Novel Artificial Neural Network and an Improved Particle Swarm Optimization used in Splice Site Prediction", Journal of Applied and Computational Mathematics, 3.

[61] Nassa, T. Singh, S. and Goel, N., (2013). "Splice site detection in DNA sequences using probabilistic neural network", International Journal of Computer Applications(IJCA), 76: 1-4.

[62] Lu, Y. Qu, W. Shan, G. and Zhang, C., (2015). "DELTA: A Distal Enhancer Locating Tool Based on AdaBoost Algorithm and Shape Features of Chromatin Modifications", PLoS ONE, 10: e0130622.

[63] Hastie, T. Tibshirani, R. and Friedman, J., (2009). The elements of statistical learning: data mining, inference, and prediction: Springer-Verlag New York.

[64] Zhong, L. Wang, J.T.L. Wen, D. Aris, V. Soteropoulos, P. and Shapiro, B.A., (2013). "Effective Classification of MicroRNA Precursors Using Feature Mining and AdaBoost Algorithms", OMICS A Journal of Integrative Biology, 17: 486-493.

[65] Xie, X. Wu, S. Lam, K.-M. and Yan, H., (2006). "PromoterExplorer: an effective promoter identification method based on the AdaBoost algorithm", Bioinformatics, 22: 2722–2728.

[66] Pashaei, E. Ozen, M. and Aydin, N., (2016). "Biomarker Discovery based on BBHA and AdaboostM1 on Microarray Data for Cancer Classification", IEEE 38th Annual International Conference of the Engineering in Medicine and Biology Society (EMBC) Orlando,USA: IEEE: 3080-3083.

[67] Saeys, Y. Degroeve, S. Aeyels, D. Rouze, P. and Peer, Y., (2004). "Feature selection for splice site prediction: A new method using EDA-based feature ranking", BMC Bioinformatics, 5.

[68] Saeys, Y. Degroeve, S. Aeyels, D. Van, P.D. and Rouze, P., (2003). "Fast feature selection using a simple estimation of distribution algorithm: A case study on splice ste prediction", Bioinformatics, 19: 179-188.

[69] Svetnik, V. Liaw, A. and Tong, C., (2004). "Variable Selection in Random Forest with Application to Quantitative Structure-Activity Relationship", USA: Springer-Verlag.

[70] Genuera, R. Poggi, J.M. and Malotc, C.T., (2010). "Variable selection using Random Forests", Pattern Recognition Letters,Elsevier, 31: 2225-2236.

[71] Han, L. Embrechts, M.J. Szymanski, B. Sternickel, K. and Ross, A., (2006). "Random Forests Feature Selection with Kernel Partial Least Squares: Detecting Ischemia from Magneto Cardiograms",European Symposium on Artificial Neural Networks Burges, Belgium: 221-226.

[72] Reif, D.M. Motsinger, A.A. McKinney, B.A. Crowe, J.E. and Moore, J.H., (2006). "Feature Selection using a Random Forests Classifier for the Integrated Analysis of Multiple Data Types",Symposium on Computational Intelligence and Bioinformatics and Computational Biology (CIBCB'06) Toronto: IEEE: 1-8.

[73] Slavkov, I. Zenko, B. and Dzeroski, S., (2009). "Evaluation Method for Feature Rankings and their Aggregations for Biomarker Discovery",3rd International Workshop on Machine Learning in Systems Biology (MLSB) Ljubljana, Slovenia: 122-135.

[74] Kocev, D. Slavkov, I. and Dzeroski, S., (2013). "Feature ranking for multi-label classication using predictive clustering trees".

[75] Pollastro, P. and Rampone, S., (2002). "HS3D, a dataset of Homo sapies splice site regions, and its extraction procedure from a major public database", International journal of Modern Physics, C13: 1105-1117.

[76] Xia, X., (2012). "Position Weight Matrix, Gibbs Sampler, and the Associated Significance Tests in Motif characterization and Prediction", Scientifica (Cairo), 2012: 1-15.

[77] Hertz, G.Z. and Stormo, G.D., (1999). "Identifying DNA and protein patterns with statistically significant alignments of multiple sequences", Bioinformatics, 15: 563-577.

[78] Meher, P.K. Sahu, T.K. Rao, A.R. and Wahi, S.D., (2016). "Identifcation of donor splice sites using support vector machine: a computational approach based on positional, compositional and dependency features", Algorithms for Molecular Biology, 11.

[79] Ang, J.C. Mirzal, A. Haron, H. and Hamed, H.N.A., (2016). "Supervised, Unsupervised, and Semi-Supervised Feature Selection: A Review on Gene Selection", IEEE/ACM Transaction on Computational Biology and Bioinformatics, 13: 971-989.

[80] Kumari, B. and Swarnkar, T., (2011). "Filter versus Wrapper Feature Subset Selection in Large Dimensionality Micro array: A Review ", International Journal of Computer Science and Information Technologies (IJCSIT), 2: 1048-1053.

[81] Hira, Z.M. and Gillies, D.F., (2015). "A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data", Advances in Bioinformatics, 2015.

[82] Filimon, A., (2011). Hedge fund fraud prediction using classication algorithms, ed^eds. Science in Applied Mathematics. Merlin: University of Zurich.

[83] Bindewald, E. and Shapiro, B.A., (2006). "RNA secondary structure prediction from sequence alignments using a network of k-nearest neighbor classifiers", RNA, 12: 342–352.

[84] Freund, Y. and Schapire, R.E., (1999). "A Short Introduction to Boosting", Journal of Japanese Society for Artificial Intelligence, 14: 771-780.

[85] Freund, Y. and Schapire, R.E., (1996). Experiments with a New Boosting Algorithm, L. Saitta, ed. Proceedings of the Thirteenth International Conference on Machine Learning (ICML 1996). Morgan Kaufmann.

[86] Zhang, H. and Gu, C., (2006). Support Vector Machines versus Boosting, ed^eds. USA: University of California, Berkeley, 1-19.

[87] Mease, D. and Wyner, A., (2008). "Evidence Contrary to the Statistical View of Boosting", Journal of Machine Learning Research, 9: 131-156.

[88] Cortes, C. and Vapnik, V., (1995). "Support vector network", Mach. Learn, 20: 273–297.

[89] Haykin, S., (1999). Neural Networks: a comprehensive foundation, Upper Saddle River: Prentic Hall.

[90] Breiman, L., (2001). "Random Forest", Mchine Learning, 45: 5-32.

[91] Statnikov, A. Wang, L. and Aliferis, C.F., (2008). "A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification", BMC Bioinformatics, 9.

[92] Goel, N. Singh, S. and Aseri, T.C., (2013). "A comparative analysis of soft computing techniques for gene prediction", Analytical Biochemistry, 438: 14-21.

[93] Gray, K.R. Aljabar, P. Heckemann, R.A. and Rueckert, D., (2012). "Random forest-based similarity measures for multi-modal cassification of Alzheimer's disease", NeuroImage 65: 167–175.

[94] Saito, T. and Rehmsmeier, M., (2015). "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets", PLoS ONE, 10: e0118432.

[95] Lin, W.J. and Che, J.J., (2012). "Class-imbalanced classifiers for high-dimensional data", Briefings in Bioinformatics, 14: 13-26.

[96] Ganganwar, V., (2012). "An overview of classification algorithms for imbalanced datasets", International Journal of Emerging Technology and Advanced Engineering(IJETAE), 2: 42-47.

[97] Longadge, R. Dongre, S.S. and Malik, L., (2013). "Class Imbalance Problem in Data Mining: Review", International Journal of Computer Science and Network (IJCSN), 2: 83-87.

[98] Saito, T. and Rehmsmeier, M., (2015). "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets", PLoS ONE, 10.

[99] Liu, H. and Motoda, H., (1998). Feature Selection for Knowlegde Discovery and Data Mining london: Kluwer Academic Publisher.

CURRICULUM VITAE

PERSONAL INFORMATION

Name Surname : Elham PASHAEI

Date of birth and place : 1985, IRAN Urmia

Foreign Languages : Turkish, English

E-mail : Elham.pashaei@std.yildiz.edu.tr

EDUCATION

Degree	Department	University	Date of Graduation
Master	Computer Engineering	Azad Qazvin University	2012
Undergraduate	Computer Engineering	Azad Khoy University	2008
High School	Mathematics & physics	---	2003

PUBLISHMENTS

Papers

1. Pashaei, E. Ozen, M., and Aydin, N., (2017). "Splice site identification in the human genome using random forest", Health and Technology, 7: 141-152.

2. Pashaei, E. and Aydin, N., (2017). "Prediction of Human splice sites using AdaBoost with efficient DNA encoding approaches", Frontiers of Information Technology & Electronic Engineering, (under review).

Conference Papers

1. Pashaei, E. Ozen, M., and Aydin, N., (2016). "Random Forest in Splice Site Prediction of Human Genome", XIV Mediterranean Conference on Medical and Biological Engineering and Computing Paphos, Cyprus: Springer International Publishing: 512-517.

2. Pashaei, E. Ozen, M., and Aydin, N., (2016). "Splice Sites Prediction of Human Genome using AdaBoost", IEEE International Conference on Biomedical and Health Informatics (BHI 2016) Las Vegas, USA: 300-303.

3. Pashaei, E. Yilmaz, A. Ozen, M. and Aydin, N., (2016). "A Novel Method for Splice Sites Prediction Using Sequence Component and Hidden Markov Model", 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) Florida, USA: 3076 - 3079.

4. Pashaei, E. Yilmaz, A. and Aydin, N., (2016). "A Combined SVM and Markov Model approach for Splice Site Identification", 6th IEEE International Conference on Computer and Knowledge Engineering (ICCKE2016) Mashhad, Iran: 200-2004.

5. Pashaei, E. Yilmaz, A. Ozen, M. and Aydin, N., (2016). "Prediction of splice site using AdaBoost with a new sequence encoding approach", 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC) Budapest, Hungary: 3853-3858.