Dissertation

A Study on the Effect of Feature Selection against Categorical and Numerical Features in Fixed-length DNA Sequence Classification

Graduate School of
Natural Science & Technology
Kanazawa University

Division of Electrical Engineering
and Computer Science

Student ID No.: 1424042019
Name: Phan Dau
Chief advisor: Professor Kenji Satou
Date of Submission: June 30th, 2017

Abstract

DNA sequence classification can be defined as a process that classifies unknown DNA sequences into labelled classes. DNA sequence classification methods are categorized in three-fold. The key steps in the first category, distance-based methods, include defining distance functions to weigh the similarity between two DNA sequences, then applying present classification algorithms like k-nearest neighbor algorithm. For second group, feature-based methods, DNA sequences, firstly, are encoded into numerical feature vectors, then conventional algorithms like support vector machines are applied to classify DNA sequences. The last group are model-based methods. They are related to applying hidden Markov model or employing statistical models to address the problem of DNA sequence classification.

In order to solve this problem, there were several studies that used numerical features such as k-mer to classify DNA sequences. There were also researches using categorical features. However, the combination of numerical features and categorical features have not been studied. K-mer frequency, for example, is well-known numerical feature but position-independent. Sequences with different lengths are encoded into the same size of feature vectors, but positional information is ignored. In contrast, categorical feature such as subsequence at a position are position-specific. It provides positional information, but it loses quantitative information. Therefore, it can be expected that the combination of quantitative and positional information could help improve the classification performance. Moreover, it is not clear that whether or not a feature selection algorithm could be effective on the union of the numerical and categorical features of DNA sequences. By utilizing the combination of numerical features like k-mers and categorical features like subsequences starting at a specific position of given DNA sequence, in this research, we developed a simple but efficient model for improving the performance of fixed-length DNA sequence classification.

The proposed model was evaluated on six benchmark datasets. Promoter and Splice dataset were downloaded from UCI machine learning repository. Human, worm and fly datasets are benchmark datasets which were developed by Guo *et al.* in 2014, and yeast dataset is the benchmark dataset developed by Chen *et al.* in 2015.

The performance results of our model were comparable or better than those of active algorithms. The most noticeable thing is that our method reached the accuracy of 100% on two datasets: Promoter and yeast datasets. What more is that by performing feature selection on numerical and categorical features, we could also discover which group of features are more effective on which dataset.

Keywords: Sequence classification, Numerical and categorical features, Feature selection

Acknowledgements

After three years studying at Kanazawa University, this thesis is the best award for me. I believe that the completion of this work would have been immeasurably more difficult if without the help and support from the kind people. Therefore, I wish to take this opportunity to gratefully acknowledge the assistance and contribution of everyone who made my work accomplish fully.

First of all, I am deeply thankful to my supervisor, Professor Kanji Satou for his enthusiasm, encouragement and support. I thank him for providing me a chance to do research at the Bioinformatics Laboratory, Kanazawa University. His advice on both research and daily life have been precious for me.

I am earnestly grateful to all committee members, Professor Kenji Satou, Professor Takeshi Fukuma, Associate Professor Yoichi Yamada, Associate Professor Hidetaka Nambo, and Associate Professor Makiko Kakikawa for reading my thesis and giving brilliant comments and suggestions.

Thanks are also due to the Board of Kien Giang 120 Project under Mekong 1000 Project for offering me the scholarship. This scholarship has played a key role in achieving my educational dreams.

I would like to thank Kanazawa University for providing me a chance to take the PhD course here. My great thanks are extended to the staff of Kanazawa University who are friendly, polite and enthusiastic. It is strongly believed that my studying here would be more difficult if I did not get assistance and help from them.

I deeply thank to all colleagues at Kien Giang Teacher's Training College, especially to the Managing Board of College who allowed me take the PhD course. Moreover, their encouragement and help enabled me to overcome difficulties and challenges.

My sincere thanks also goes to Vietnamese students' Union at Kanazawa University (Vietkindai) for what the Union has done for us. It makes me feel Kanazawa land as my second home where there are so many friends who I can share my feelings when I feel sad or happy. I would like to show my deep gratitude to two Japanese uncles,

Yukio Abe and Mi Tetuo for their support and contribution to the success of the Union.

My deep thanks also go to my friends at the Bioinformatics Laboratory, Kanazawa University for wonderful moments we have worked together. I wish to acknowledge the help provided by all my labmates, Vu Anh, Sergey, Duc Luu, Ngoc Giang, Rosi, Bahriddin, Reza, Bedy and Mera, who shared their expertise and experiences with me, and provided me valuable feedbacks and suggestions.

Last but not the least, I owe my deepest gratitude to my parents, parents-in-law, brothers, sisters, especially my wife and my daughter. They have always stood by me in all situations. Most importantly, I am forever indebted to my wife and my daughter for their love, understanding and encouragement. They have been waiting my return with much patience for three years to become a complete family.

Thank you so much!

Table of Contents

List of Figures

List of Tables

Chapter 1 : Introduction

We begin in section 1.1 by introducing a research context of the thesis and various types of sequence classification methods. These include distance-based methods, feature-based methods and model-based classification. Objectives of the thesis are mentioned in section 1.2. We describe the contributions of the thesis in section 1.3. Lastly, in section 1.4 we show the organization of the thesis.

1.1. Research context

In recent years, biological data have been generated at a tremendous rate. According to [1], the number of DNA sequences contained in GenBank repository increased dramatically from 116,461,672 to 181,336,445 between February 2010 and February 2015 (as shown in Figure 1.1). The sequences in UniProt doubled during the period of just one year, from 40.4 (June 2013) to 80.7 (August 2014) million [2]. Analysis and interpretation of these data are two of the most crucial tasks in bioinformatics, and classification and prediction methods are key techniques to address such tasks.

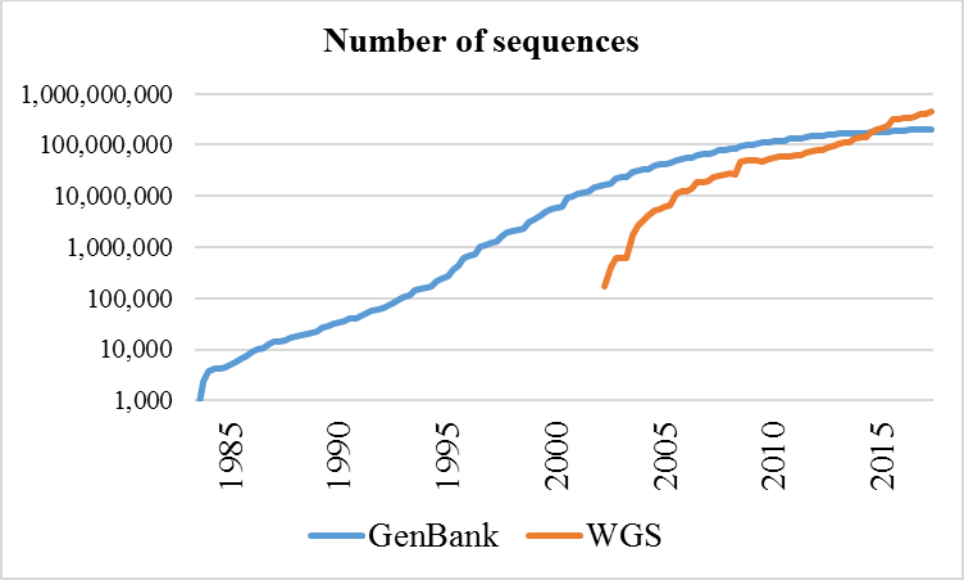


Figure 1.1. The number of sequences in GenBank and WGS

As summarized by Xing *et al*. [3], there were three main groups of the DNA sequence classification approaches. The first class includes methods that firstly define distance functions to compute the similarity between two sequences. After that, some of the current learning algorithms like k-nearest neighbor are applied. The second category is feature-based methods. Before employing conventional algorithms such as decision trees and artificial neural networks to classify DNA sequences, these sequences need to be encoded into feature vectors. So as to enhance the performance, feature selection plays a key role in this type of methods. The last type is a group of methods like Markov Models and statistical algorithms used to perform sequence classification.

With regard to the first category, in the research of by Borozan *et al*. [4] in 2015, they exploited the complementarity of alignment-free and alignment-based similarity to classify biological sequences. They used five different sequence similarity measures: three out of five measures were alignment-free and the other two belonged to the second category of measures, which revealed that their model outperformed previous models. In 2014, Chen *et al.* [5] also tackled the problem of categorical data in a typical distance-based manner. They defined four weighted functions for categorical features, two of them named as simple matching coefficient measures with global weights ($WSMC_{global}$) and the other two named as simple matching coefficient measures with local weights ($WSMC_{local}$), then applied these functions to formulate new nearest neighbor classification algorithms. The classifiers were evaluated by using real datasets and biological datasets. The results showed that their proposed classifiers outperformed the traditional methods.

Moving to the second class, the application of feature selection technique and feature-based method to classify protein sequence data was carried out by Iqbal *et al* [6] in 2014. The experimental results of their research showed that their model significantly improved in terms of accuracy, sensitivity, specificity, F-measure, and other performance measure metrics. In the study of Weitschek *et al*. [7] in 2015, they used the combination of alignment-free approaches and rule-based classifiers so as to classify biological sequences. At first, the biological sequences were converted into numerical feature vectors with alignment-free techniques, then rule-based classifiers were applied in order to assign them to their taxa.

The study about classifying occupancy, acetylation, and methylation of nucleosomes was carried out by Pham *et al*. [8]. Their method was also a kind of feature-based classification, which converted sequences into numerical feature vectors, then applied a conventional classification method. They adopted SVM with RBF kernel, and feature vectors were k-mer based vectors with a variety of window sizes (k =3, 4, 5, 6, etc.). Using ten datasets collected by Pokholok *et al*. [9], they gained a high prediction accuracy. To get better performance results, a technique termed feature selection was used by Higashihara *et al*. [10] to solve this problem. In this research, the importance of features was first measured by MeanDecreaseGini

value computed through training and prediction by random forest, then features were ranked as the order from the most to least importance. Exploiting feature selection along feature ranking, they achieved slight improvements in prediction accuracy. What is more, by searching neighbors of the best feature subset, accuracy of prediction improved further.

Although the active researches on sequence classification above, numerical and categorical features were separately studied until now. Since the numerical features like k-mer are typically position-independent and categorical features like nucleotide at a position are position-specific, we can expect that these two types of features could contribute to the classification performance in a complementary manner. In addition, it is still unclear how effective a feature selection algorithm is against the union of numerical and categorical features of sequence. In this study, we propose an effective framework for improving fixed-length DNA sequence classification by using the combination of categorical features (i.e. subsequence at a position like “A”, “AG”, etc.) and numerical features (i.e. k-mer frequency). By conducting feature selection on this mixture of features, we could also find which type of features are more effective in each dataset.

1.2. Objectives

The major target of our thesis is that we develop an effective model to address the problem for classifying fixed-length DNA sequences. The specific objectives are as follows.

Applying a proposed model to classify promoter sequences

In genetics, gene expression is the series of biological actions which is related to using DNA of the gene to synthesize the functional product. These functional products are often proteins which have important roles in organisms, including catalyzing reactions, transporting molecules like oxygen, keeping organisms healthy and transmitting messages. The process of gene expression has two key stages: transcription and translation. The first stage of gene expression is transcription, which engages in copying information from a gene to produce an RNA molecule (especially

mRNA). This step is performed by enzymes called RNA polymerases, which use a DNA strand as a template to form an RNA. Translation is the second stage of gene expression. It is the process of translating the chain of a messenger RNA molecule to a chain of amino acids, polypeptide. The polypeptide then folds into an active proteins and fulfils its functions in the cell. The diagram in Figure 1.2 shows a process of gene expression in a eukaryote.

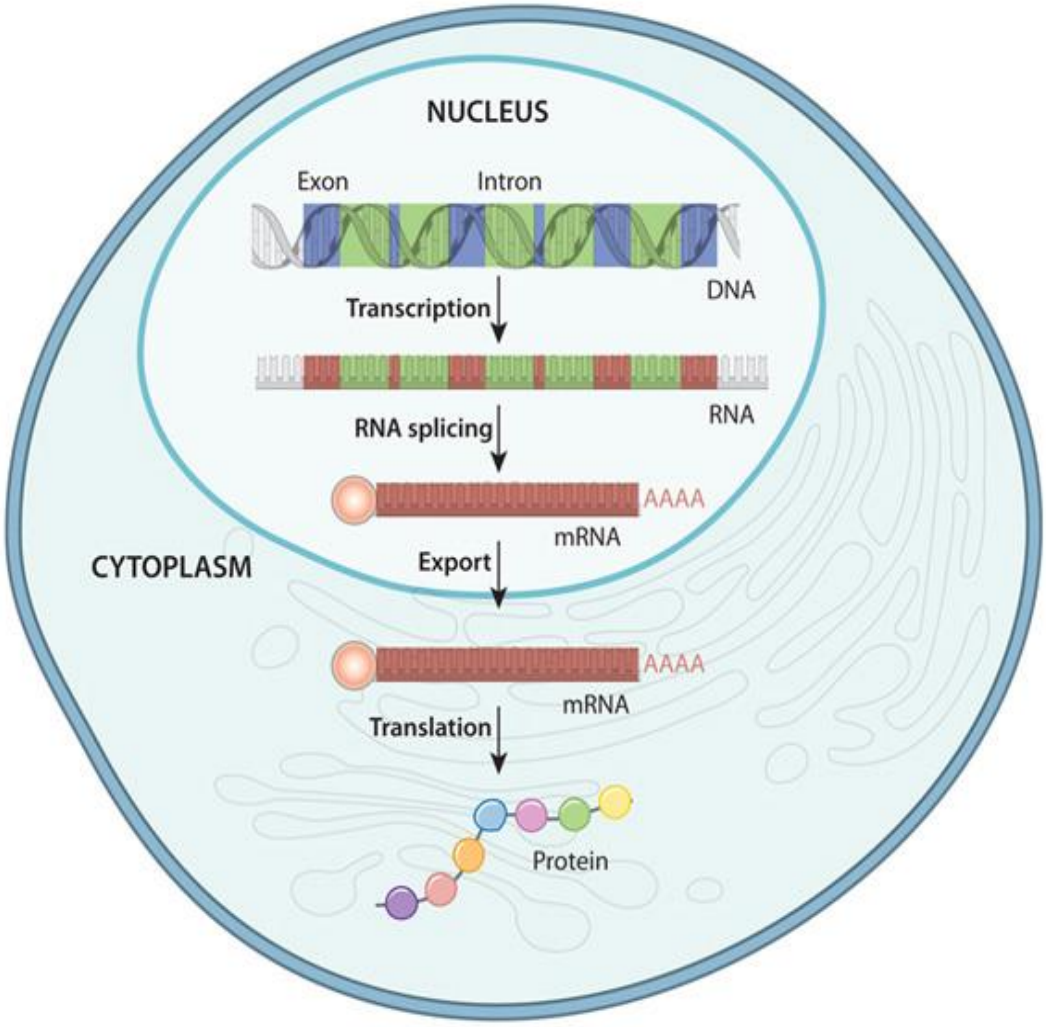


Figure 1.2. The process of gene expression. (Source *[11]*).

A promoter is the part of DNA sequence which are sited directly upstream of the start site of transcription. Figure 1.3 shows the position of the promoter on a DNA sequence.

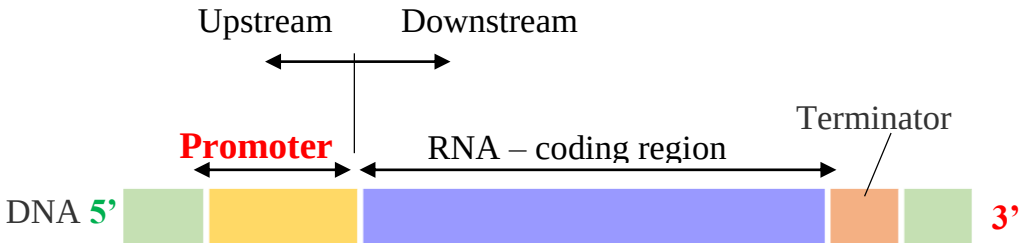


Figure 1.3. Promoter position in DNA sequence

The most important step in the process of transcription is to determine where is a gene or where is the transcription start site. Promoter identification can help locate the position of gene and then analyze the process of gene expression. Therefore, promoter prediction and promoter classification are two considering problems in the field of bioinformatics, and classifying promoter sequences is the first objective of our research.

Applying a proposed model to classify splice sequences

During transcription process, the pre-mRNA is stranscribed from a DNA of gene. This pre-mRNA must be processed in order to become a mature messenger RNA (mRNA) that can be translate into a protein. The step of processing the pre-mRNA to become mature mRNA is called as RNA splicing, and it is carried out by spliceosome. Since pre-mRNAs include introns and exons, hence introns must be removed by the process of splicing to form mature mRNAs. After the process of RNA splicing, mature mRNAs are translated into proteins. Two types of splice junctions are exon-intron (EI) and intron-exon (IE) junctions. The first one is called a donor site that usually contains dinucleotides GT. However, the later is named as an acceptor site which often includes dinucleotides AG [12]. Figure 1.4 illustrates the positions of them in the DNA sequence.

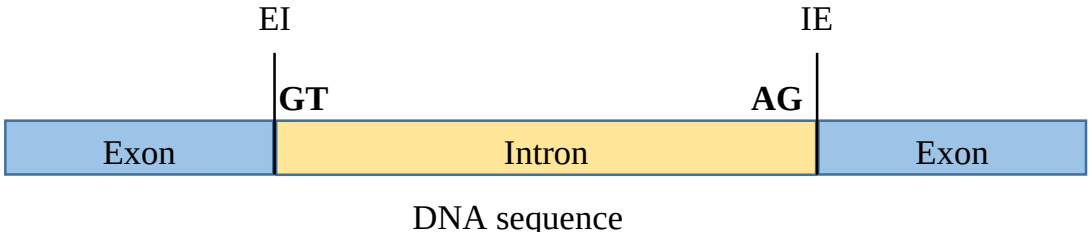


Figure 1.4. Two types of splice junctions in a DNA sequence.

In eukaryotes, the first important works for predicting gene is to predict splice junctions. Therefore, developing an effective model for accurately predicting splice junctions is an attractive work, and it is also the second goal of our research.

Applying a proposed model to classify nucleosomal sequences

In eukaryotes, one of the fundamental parts forming chromatin includes nucleosome. Every nucleosome is composed of a segment of roughly 147 base pairs (bp) which is called a nucleosome core particle being covered stiffly around a histone octamer [13]. In order to form the histone octamer, eight histone proteins are used. They include two H2As, two H2Bs, two H3s and two H4s as shown in Figure 1.5. Two nucleosome core particles are connected each other by a linker DNA sequence. Several researches indicated that nucleosome core particle played crucial roles in biological processes like DNA replication and DNA repair [14], [15], [16], [17], [18]. Therefore, predicting nucleosome positioning sequences (or nucleosomal sequences) is fundamentally important in bioinformatics. This problem is also thirdly addressed in our research.

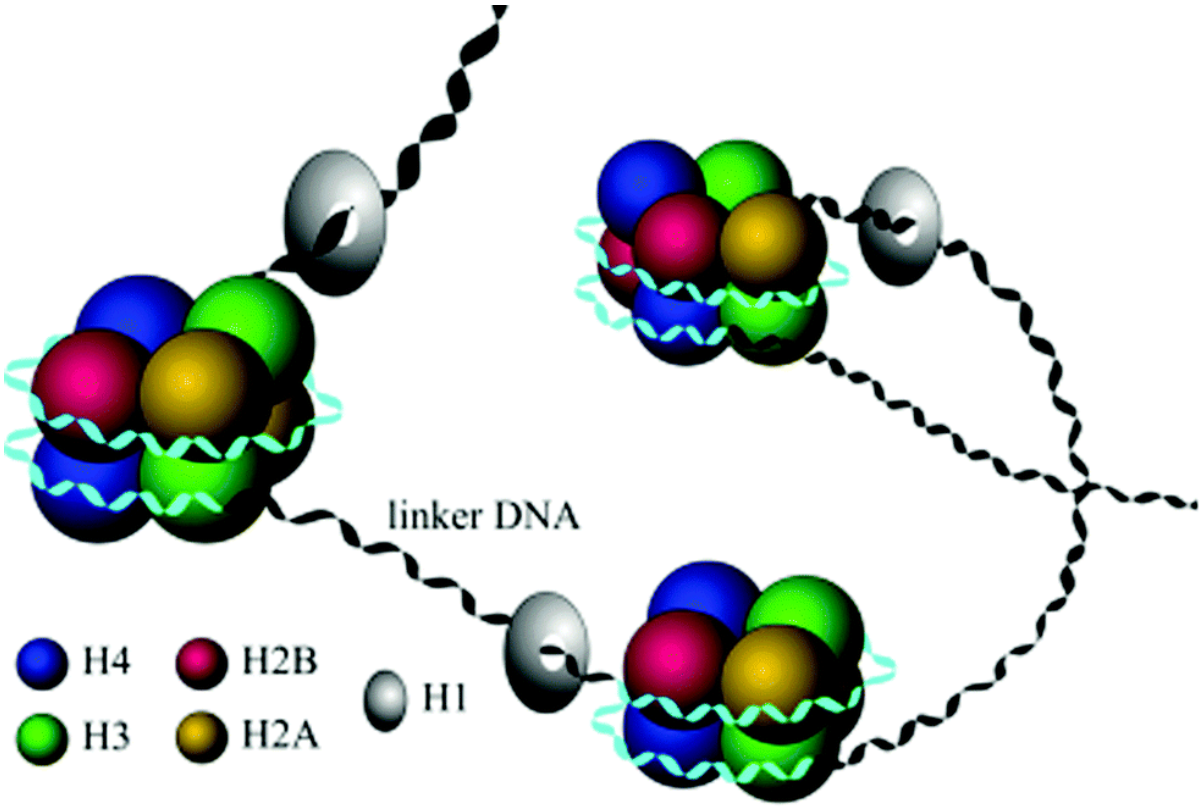


Figure 1.5. The structure of nucleosome. (Source *[13]*)

1.3. Contributions

In order to classify DNA sequences, k-mer frequency, quantitative information, is commonly used since it can convert sequences with different lengths into the same size of feature vectors. However, positional information is lost in k-mer. For fixed-length sequence, it is possible to use subsequence itself as categorical value. It keeps positional information, however, quantitative information is not available. The primary goal of our dissertation is to provide a framework for classifying fixed-length DNA sequences by using both positional and quantitative information. The key contributions of present thesis can be summarized as follows.

An effective framework for improving fixed-length DNA sequence classification

There were researches that employed numerical features for DNA sequence classification. There were also studies using categorical features. However, until now there is no research that utilized the combination of categorical features and numerical feature in one model. Therefore, we developed a simple but effective model for classifying fixed-length DNA sequences by combining numerical features (quantitative information) with categorical features (positional information) in one model. Feature selection was also used so as to improve the prediction accuracy.

Applications to various biological datasets

Our framework was applied to three different types of DNA sequence datasets: a splice junction sequence dataset, a promoter sequence dataset and four nucleosome positioning datasets. Through the performance evaluation on six datasets of fixed-length DNA sequences, our algorithm based on the above idea achieved comparable or higher results than other advanced algorithms.

Discovery of effective features

By conducting feature selection on the combination of numerical and categorical features, we could also find which type of features are more effective in each dataset. For Promoter and Splice datasets, categorical features, especially 2CAT features, are more effective than other numerical features. Whereas several numerical

features in k-mers are so important than categorical features on nucleosome positioning datasets.

1.4. Organization

This thesis is divided into five chapters. It is organized as follows.

The current one is Chapter 1 that introduces the research context, objectives, contributions and the organization of this thesis.

Chapter 2 talks about related works. At first, we review some models for predicting promoter, splice site and nucleosome positioning sequences. Next, we highlight some well-known algorithms as well as evaluation measures that have been used in bioinformatics.

Chapter 3 describes our model in detail. We, firstly, provide a brief description of promoter, splice site and nucleosome positioning datasets. Then, we explain the general model, algorithms and evaluation measures.

Chapter 4 focuses on analyzing and discussing experimental Results. We also conduct evaluation and make further comparisons with state-of-the-art models.

Chapter 5 summarizes the thesis, highlights the achievements of our work and suggests some future works.

Chapter 2 : Related Works

In this chapter, we, firstly, review models related to our research. Splice site prediction review is done section 2.1. Then, we do promoter prediction review in section 2.2 and nucleosome positioning prediction in the next section. Several popular learning machine algorithms used in bioinformatics are shown in section 2.4. Next, we summarize some feature selection methods in section 2.5. Lastly, we describe well-accepted classification evaluations in section 2.6.

2.1. Splice site prediction review

There are many important algorithms like Artificial neural networks, Bayesian classifiers and SVMs that have been employed to solve splice site prediction problems [19], [20], [21], [22]. Zhang *et al*. [23] summarized some well-known models used in prediction of splice site in the period of between 1997 and 2003 as shown Table 2.1.

Table 2.1. Several popular models for detecting splice sites from 1997 to 2003 [23].

Models	References
Statistical Models:	
– Logit linear algorithm	Brendel and Kleffe, 1998 [24]
– Quadratic discriminant analysis	Zhang and Luo, 2003 [25]
– Naïve Bayes classifier	Degroeve *et al*., 2002 [26]
Decision trees:	
– Maximal dependence decomposition	Burge *et al*., 1997 [27]
– MDD with Markov model	Pertea *et al*., 2001 [28]
– C 4.5 induction tree	Patterson *et al*., 2002 [29]
Artificial neural networks:	
– Percepton	Weber, 2001 [30]
– Multi-layer Backpropagation	Reese *et al*., 1997 [31]; Sonnenburg *et al*., 2002 [32]
SVMs:	
– Linear kernels	Degroeve *et al*., 2002 [26]
– Polynomial kernels	Patterson *et al*., 2002 [29]

Apart from these researches, there have been a number of other studies on the prediction of splice sites as well. The study of using support vector machine for accurately predicting splice sites was introduced by Sonnenburg *et al.* [33] in 2007. They applied the so-called *weighted degree* kernel to solve the problem of splice sites recognition, which turned out well suitable for their work. They conducted several experiments on splice site genomes: Arabidopsis thaliana, Danio rerio,

Caenorhabditis elegans, Drosophila melanogaster, and Homo sapiens. The results revealed that their model accurately identified splice sites in these genomes. Their method achieved higher performance than past methods consisting of GeneSplicer [28] and SpliceMachine [34]. The splice site prediction tool using their method was also provided.

In 2008, Baten *et al.* [35] introduced the research on identification of splice site by exploiting informative features and employing attribute selection. They developed the algorithm using the combination of informative features with support vector machine. They also applied a feature selection method to eliminate unimportant features. They carried out the experiments on NN269 dataset. The results showed that their method outperformed the previous methods not only prediction accuracy but also training time.

By using short sequence motifs, Meher *et al*. [36] in 2014 released the statistical method for predicting donor splice sites. Their approach was used to predicted these splice sites but their sequences were not converted into numerical features. The main idea of the approach exploited dinucleotide association. The method was applied to predict human genome and the performance was compared with the common methods used in researches [27], [37], [38]. Their model achieved equal or higher accuracy than active methods. They also provided a user friendly website using this model.

Two years later, Meher *et al*. [39] also proposed another algorithm for solving the above problem. In their research, splice site DNA sequences, firstly, were converted into numerical features based on not only neighboring but also non-neighboring dinucleotide dependencies. Then, they employed three different learning machine algorithms to predict this type of DNA sequences. The model was tested on H3SD, NN269 datasets, and was further evaluated on the independent dataset. By using the independent dataset, their model obtained better accuracies compared to the previous methods such as NNsplice, MEM, MDD, WMM, MM1, FSPLICE (accessible at http://www.softberry.com), GeneID [31] *and ASSP* [40]. An online donor splice site prediction server (named as PreDOSS) was also provided.

At the same year (2016), Meher *et al*. [41] introduced another approach for predicting donor splice sites. Firstly, each sequence was encoded into three distinct groups of variables. They were positional, compositional and dependency variables. The positional variables and the scores of WMM were similar. The dependency variables and the scores WAM were similar. The compositional variables were the union of 2-mer, 3-mer and 4-mer frequency. Therefore, there were 344 variables for each sequence. Then, they conducted feature selection by using F-score, and just 49 out of 344 features survived from this step. Finally, these 49 variables were used as input to support vector machine for prediction. By using human, cattle, fish, worm and NN269 datasets, results showed that their method was comparable to present methods. They also developed a website HSplice for predicting this kind of genomes.

2.2. Promoter prediction review

There have been several machine learning models for predicting biological signals like promoters that carry out the transcription process. The prediction of promoters has been attracting many researchers in [42], [43], [44], [45], [46], [47], [48], [49], [50], therefore we review some notable methods to address this problem. In the year of 1990, Towell *et al*. [42] proposed a hybrid learning system named as KBANN (Knowledge-Based Artificial Neural Networks) that combined explanation-based learning and empirical learning systems. The dataset of *E. coli-2* DNA sequences was also developed to help test the KBANN model. This dataset consists of 106 sequences. Half of them contains promoters known as positive samples, while another half of sequences do not contain promoters assigned as negative samples, and each of sequences has length of 57. The experimental results on this dataset showed that the KBANN outperformed the other four methods.

Czibula *et al.* in 2012 [43] proposed the method for predicting promoter using relational association rules named as “PCRAR”. These rules are a specific class of association rules. They can characterize the relationships between variables in a dataset. The PCRAR model did not depend on any specific biological mechanisms. The strong point of PCRAR was that it could learn the distinctions between promoter sequences and non-promoter sequences without using further biological information.

They evaluated their proposed classifier and made comparison with existing methods. Their experimental results showed that their algorithm outperformed the existing models for identification of promoter sequences, which confirmed that their approach was a potential model for predicting promoter sequences.

The combination of expectation maximization clustering and support vector machine (EMSVM) for solving the above issue in bacterial DNA sequences was presented by Maleki *el at.* in 2015 [45]. There were two phases in the EMSVM algorithm. In the first stage, expectation maximization algorithm enabled to identify groups of bacterial DNA sequences that showed similar characteristics. Each of bacterial DNA sequence was clustered to different clusters with different probabilities. Then, in the second stage, the support vector machine was applied. The EMSVM model is shown in Figure 2.1.

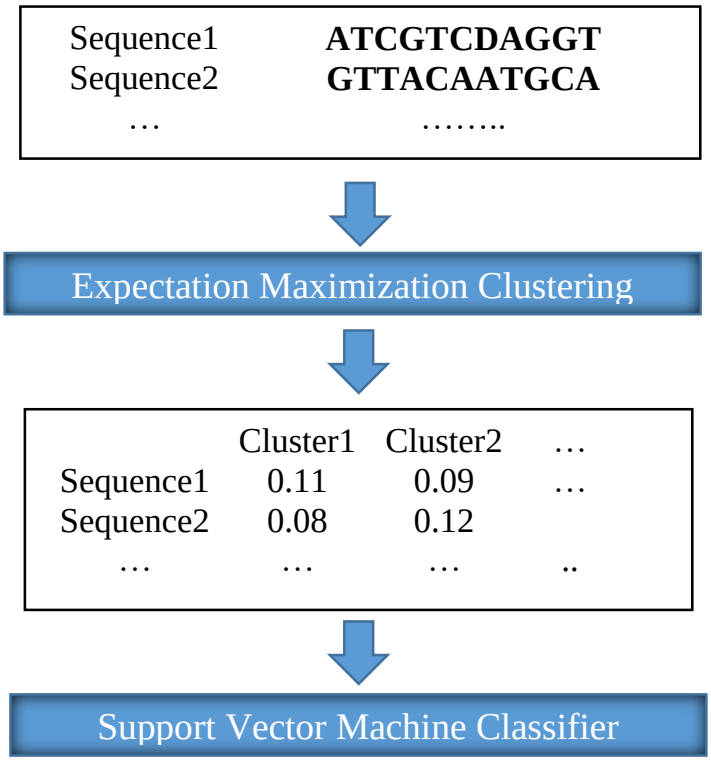


Figure 2.1. The EMSVM model proposed by Maleki *et al*. [45]

To evaluate the EMSVM method, they used the four promoter datasets of *E. coli* DNA sequences: σ24, σ32, σ38 and σ70. These datasets consist of sequences with length of 81 bases. The σ24, σ32, σ38 and σ70 contain 520, 309, 217 and 1907 promoter sequences (positive samples), respectively. Each of these datasets also

includes 2000 non-promoter sequences (negative samples). Moreover, they evaluated their proposed algorithm on the *E. coli-2* dataset as well. Four distinct evaluation metrics were used to evaluated this method. Their results demonstrated that EMSVM achieved higher performance than other methods.

Lin *et al.* [44] proposed another model, named as "iPro54-PseKNC", in 2014. In this model, firstly, promoter and non-promoter sequences were encoded into feature vectors, named as "pseudo k-tuple nucleotide composition". Next, these feature vectors were used as the inputs to support vector machine. To evaluate iPro54-PseKNC predictor, they used sigma 54 dataset and four evaluation metrics: Accuracy, specificity, sensitivity and Matthews correlation coefficient. The web server named iPro54-PseKNC was also developed.

2.3. Nucleosome positioning prediction review

A number of studies have been attempted to developed models for predicting nucleosome positioning. Herein, we summarized some methods for nucleosome positioning released recently. In 2010, Yi *et al.* [51] introduced a model for solving this problem by using transcription factor binding sites (TFBSs) and the nearest neighbor algorithm. In this research, by using an online server of MatInspector [52], both nucleosome-forming and nucleosome-inhibiting sequences were converted into feature vectors with the length of 35 transcription factor binding sites (TFBSs). Next, they conducted feature selections by using two distinct approaches to reduce the size of feature vectors. Nine important features (families of TFBSs) were identified, then nearest neighbor algorithm was applied to predict around 53000 nucleosome-forming and about 50000 nucleosome-inhibiting sequences in yeast genome. With the jackknife cross-validation test, the prediction accuracy reached 87.4%.

In the research of Guo *et al.* [13] in 2014, they proposed a predictor named as 'iNuc-PseKNC' that was applied to predict nucleosomal sequences in human, worm and fly genomes. First of all, DNA sequences in datasets were transformed to feature vectors by employing a novel formula named as "pseudo k-tuple nucleotide composition". Then, support vector machine algorithm was applied to classify DNA sequences into 2 classes: nucleosome-forming and nucleosome-inhibiting classes.

Jackknife test was also conducted to assess their method performance. They released an online server for predicting these three kinds of genomes as well.

In 2016, Tahir and Hayat [53] introduced a predictor (called “iNuc-STNC”) for prediction of nucleosome positioning in three above genomes. Nucleosome sequences were encoded into three different groups of features like 2-mer, 3-mer and split 3-mer composition. By applying support vector machine, the performance results on three benchmark datasets (human, worm and yeast) demonstrated that their model was effective on identifying nucleosome positioning sequences.

To address this problem, in early 2017 Awazu [54] developed two novel nucleosome positioning prediction models based on the combination of three kinds of features with different segment length scales. Models employed linear regression algorithm to predict nucleosomal sequences, and they were named as 3LS (three length scales) and TNS (Tri-nucleosome sequence). 3LS model achieved better results than the past methods on Homo sapiens and Drosophila melanogaster genomes, and TNS reached 100% accuracy on Saccharomyces cerevisiae genome.

2.4. Learning Machine Algorithms

There are many well-know learning algorithms used in bioinformatics. However, in this thesis, we just concentrate on several algorithms that are commonly used in splice site, promoter and nucleosome positioning prediction.

2.4.1. Artificial Neural Networks (ANNs)

ANNs are computational methods which were firstly invented in 1943 by Warren McCulloch and Walter Pitts. The key idea of an artificial neural network is simulating the human brain [55]. In the years of 1950s, with the development of the perceptron network and associated learning rule by Frank Rosenblatt, ANNs were firstly applied in the field of pattern recognition. However, now ANNs have been applied in various disciplines. In general, an artificial neural network contains three parts, known as layers. An example of an artificial neural network is shown in Figure 2.2.

The input layer has responsibility for collecting information data, signals, features from the outside environment.

The hidden layers are made of neurons which are in charge of the performance of the internal processing from a network.

The output layer includes neurons which are responsible for yielding and presenting the final network outputs.

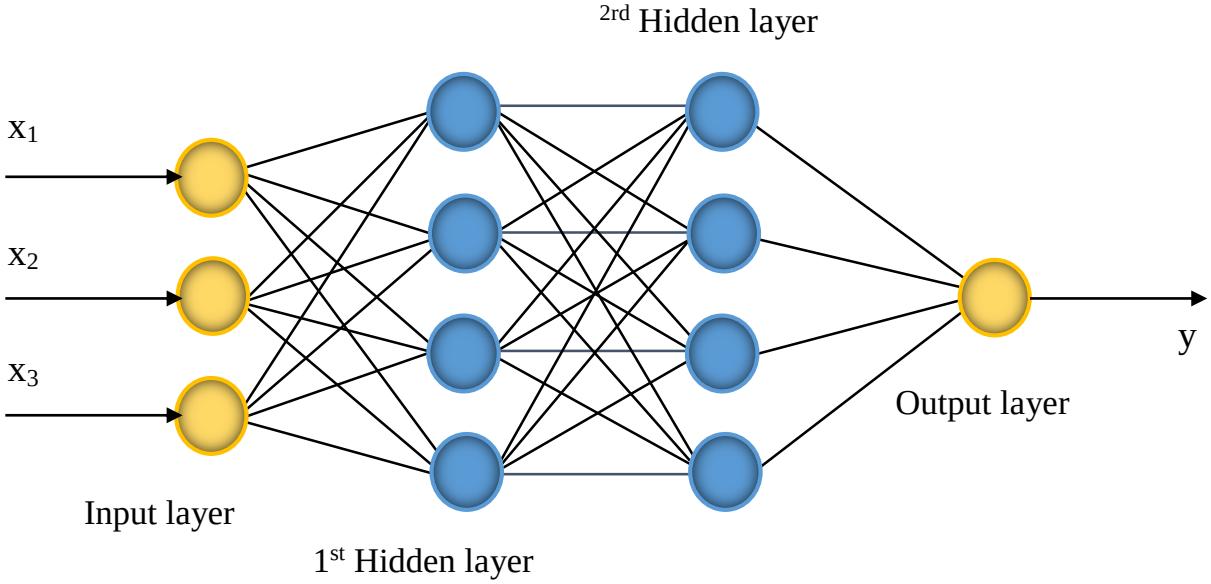


Figure 2.2. The general model of an ANN.

ANNs are constructed from layers of connected neurons. Therefore, each neuron at hidden layers and output layer receives inputs from neurons at the preceding layer and passes the output to neurons at the succeeding layer. These neurons operate in Figure 2.3 [56], [57].

$I_1, I_2, \ldots, I_n$ are the inputs for the neurons.

$w_{1j}, w_{2j}, \ldots, w_{nj}$ are the weights.

Summation = $I_1.w_{1j} + I_2.w_{2j} + \ldots + I_n.w_{nj}$

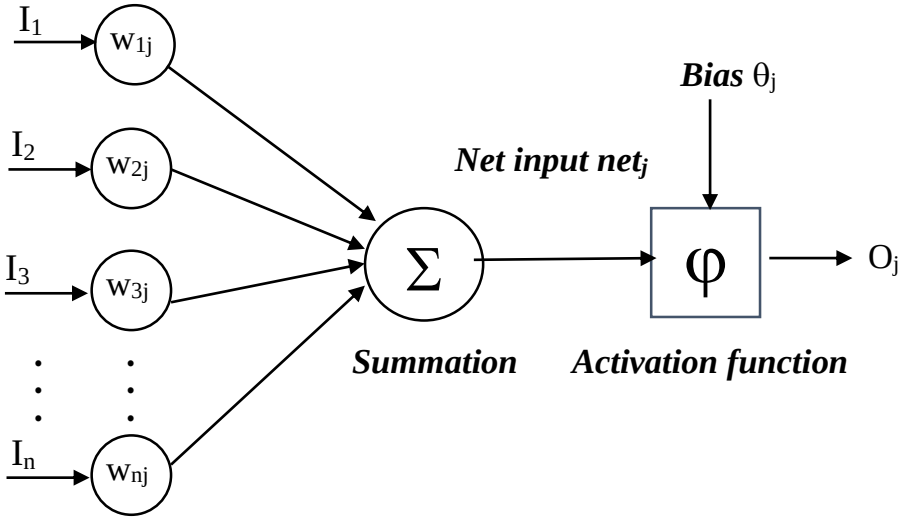


Figure 2.3. The operation of a neuron.

2.4.2. Convolutional Neural Networks (CNNs)

CNNs are one specific class of Artificial Neural Networks. They have been successfully adopted in different fields of researches like image recognition. These days, convolutional neural networks play crucial roles in the field of machine learning.

The early convolutional neural network was the LeNet Architecture (LeNet5) developed by Yamm LeCun in 1990s [58]. Figure 2.4 is the architecture of LeNet5 for recognizing characters like reading zip codes and digits, and so forth.

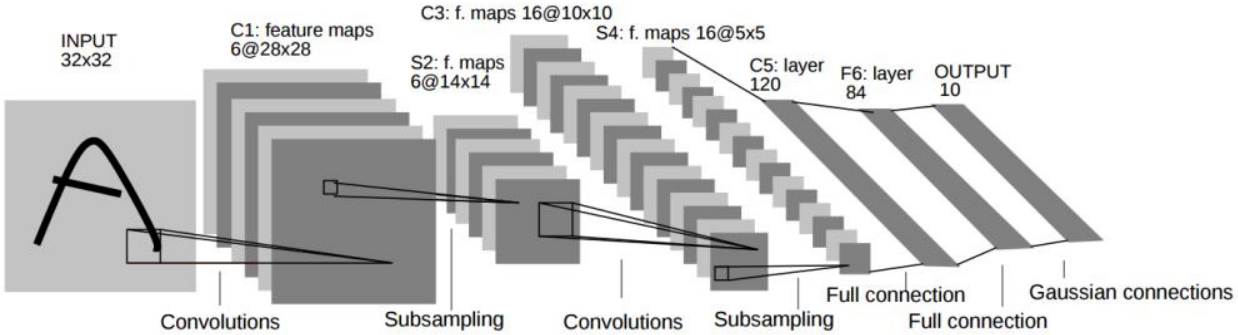


Figure 2.4. LeNet5 Architecture. (Source *[59]*)

Figure 2.5 is the structure of a CNN that has the similar architecture of the LeNet5 [58]. However, this model was used for classifying images. There are four leading operations in the convolutional neural network (shown in Figure 2.5).

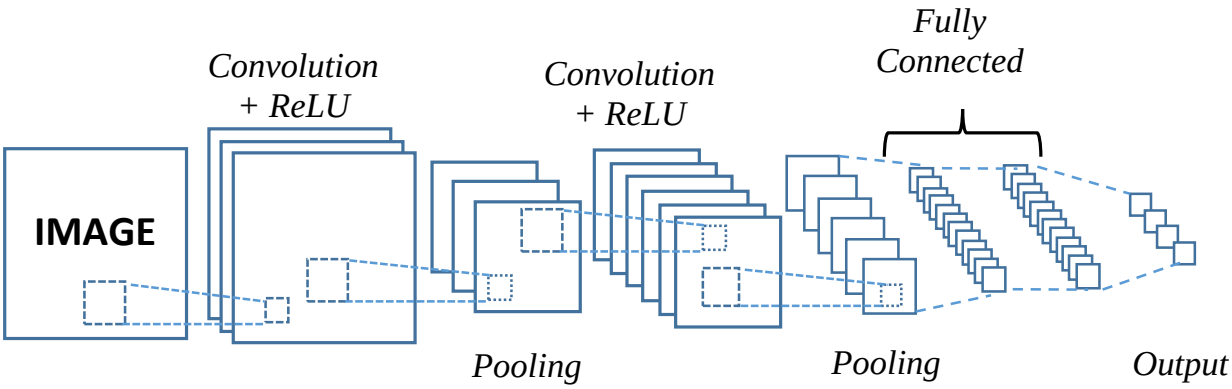


Figure 2.5. A simple structure of CNN for recognizing image.

From the first convolutional neural networks, LeNet, was invented in early 1990s. Some other convolutional neural networks have been evolved, and they are listed as follows [58].

In 2012, AlexNet was presented by Alex Kirzhevsky *et al*. This convolutional neural network had more neurons and layers than the LeNet. It was a winner in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) at that time.

In 2013, the winner in ILSVRC was ZF Net (stand for Zeiler and Fergus Net) which was developed by Matthew Zeiler and Rob Fergus.

In 2014, GoogLeNet was a winner in ILSVRC. This convolutional neural network was introduced by Szegedy *et al*. from Google.

In 2015, The winner in ILSVRC was Residual Network (ResNets) that was developed by Kaiming He *et al*.

Recently, August 2016, Huang *et al*. introduced the Densely Connected Convolutional Network (DenseNet). The DenseNet outperformed previous network architectures on five object recognition benchmark datasets.

2.4.3. Support Vector Machines (SVMs)

In machine learning, SVMs have been used for classification as well as regression. They can conduct not only on linear data but also on nonlinear data. Support vector machines were invented in mid – 1960s by Vapnik *et al*. However, until 1992, the journal article about support vector machines was firstly published,

and the authors of the paper were Vapnik and his colleagues [60]. The overview of its algorithm works as following. Support vector machine, firstly, uses a special mapping to convert the lower dimension data into a higher dimension data. Then, it finds a linear optimal hyperplane that divides tuples of one group from another group. Support vector machine searches for this hyperplane based on support vectors (the samples that are essential to detect classes) and margins. Support vector machines have been applied to various areas, consisting of digit recognition [61], image recognition [62], text classification [63], sequence classifications [64] and so forth. Some main advantages and disadvantages of support vector machine can be summarized as follows [65]:

Advantages:

- Support vector machine can work effectively in high dimensional spaces.
- Support vector machine performs much better when there is a clear margin of separation.
- Support vector machine is memory saving algorithm since it uses a subgroup of training instances called support vectors to formulate a decision function. Only these support vectors must be stored in memory in order to make decisions.

Disadvantages:

- Support vector machine does not work very well on datasets with more noise.
- Support vector machine does not directly yield probabilistic estimates for group membership.

2.4.4. Random Forests

Random Forests were invented by Breiman *et al*. in 2001 [66]. They are ensemble learning methods that can be used for classification as well as regression [60], [67]. At the training phase, random forests build a lot of decision tree classifiers so that the collection of these individual classifiers forms random forests. During classification, output class of random forests is the voting of the output classes produced by individual classifiers [60]. Figure 2.6 is an example of random forest.

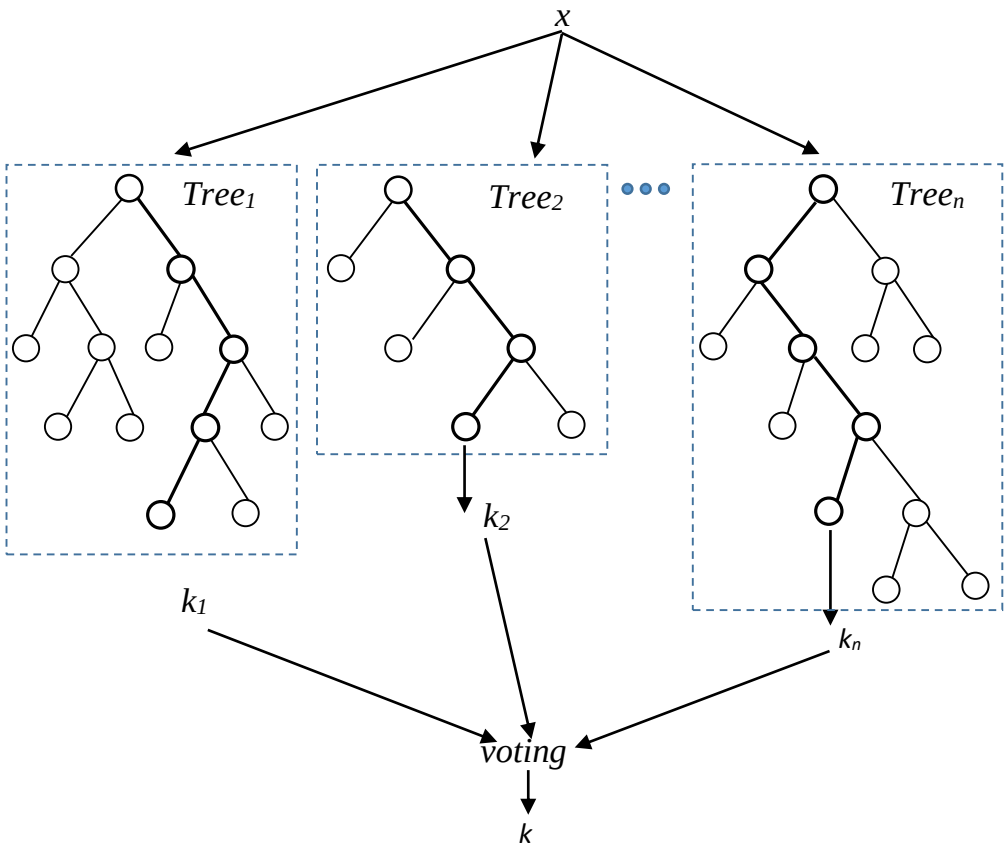


Figure 2.6. An example of random forest.

Random forests are the popular algorithms for not only classification but also regression. However, random forests have been more widely used for classification models than for regression models. Some features of Random Forests can be listed as following [68].

- In the terms of accuracy, it is not the best one among latest algorithms. However, it can work efficiently on the big datasets. Moreover, random forest has ability to handle data with a high dimension, thousands of input features.
- Random forest has a good algorithm for estimating missing data so that it can work well on the data with a large proportion of missing.
- When random forest has been generated, it can be saved for later use.
- Random forest can measure the importance of features in the classification.
- Random forest provides an effective algorithm for detection of feature interactions.

2.4.5. k-Nearest Neighbor Algorithms

In the early 1950s, the k-nearest neighbor algorithm was introduced but it was not popular. Until the 1960s, the k-nearest neighbor algorithm was commonly used since there was a significant increase in computing power [60]. The general flowchart of k-nearest neighbor algorithm (KNN) is illustrated in Figure 2.7.

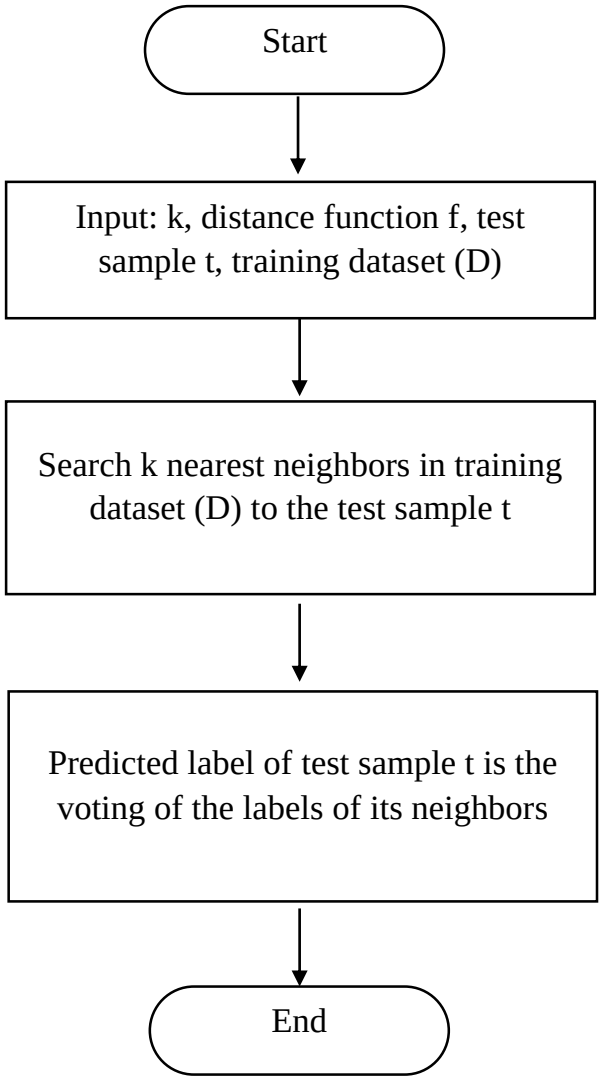


Figure 2.7. Flowchart of KNN algorithm.

There are several popular distance functions that have been applied to compute the distance of two points pp and qq in a feature space. Given pp = (pp_1, pp_2, …, pp_n) and qq = (qq_1, qq_2, …, qq_n) are two points in an n-dimension feature space.

Euclidean distance:

$$d(pp, qq) = \sqrt{\sum_{i=1}^{n}(pp_i - qq_i)^2}$$

Squared Euclidean distance:

$$d(pp, qq) = \sum_{i=1}^{n}(pp_i - qq_i)^2$$

Manhattan distance:

$$d(pp, qq) = \sum_{i=1}^{n}|pp_i - qq_i|$$

Minkowski Distance:

$$d(pp, qq) = \left(\sum_{i=1}^{n}(pp_i - qq_i)^{\lambda}\right)^{\frac{1}{\lambda}}$$

Cosine similarity:

$$\cos(pp, qq) = \frac{\sum_{i=1}^{n} pp_i \cdot qq_i}{\sqrt{\sum_{i=1}^{n} pp_i^2}\sqrt{\sum_{i=1}^{n} qq_i^2}}$$

2.5. Feature Selection

In recent decades, dataset size, the number of instances as well as of attributes, have been exploding at the great considering levels. Consequently, storing and processing these data have been becoming more challenging, and machine learning methods also have difficulties in coping with the big data. In order to implement machine learning methods more effectively, pre-processing of the data is needed. Dimensionality reduction is one of the most popular and important techniques to remove noisy and redundant features, and has become an absolutely essential step machine learning process. The general model for dimensionality reduction is shown in Figure 2.8 [69]. This step enables to speed up data mining algorithms, improve prediction accuracy. The first class of dimensionality reduction is feature extraction, and the second one is feature selection.

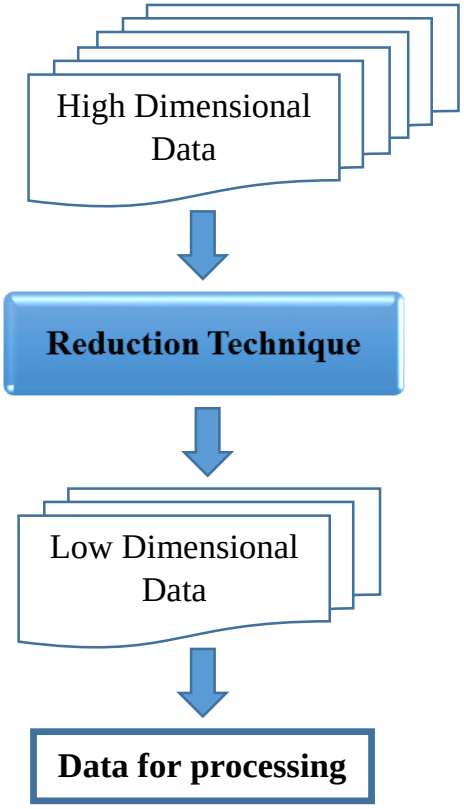


Figure 2.8. The general process of dimensionality reduction.

Feature extraction can be defined as the process of projecting the input data with a higher dimensionality into a new space with lower dimensionality.

Feature selection, however, can be defined as the process of detecting relevant features and eliminating irrelevant, redundant features. The main objectives of feature selection are threefold [70], [71], [72]: reducing computational time, improving prediction performance and understanding data deeply. There are three major categories of feature selection methods.

2.5.1. Filter Methods

Filter feature selection methods are mainly based on a number of statistical measures such as Pearson Correlation and Mutual Information, to assign different features with different scores [73]. The features with the best scores are used in building the model while others are not used for analysis. These methods are not dependent on any particular classifiers, as shown in Figure 2.9.

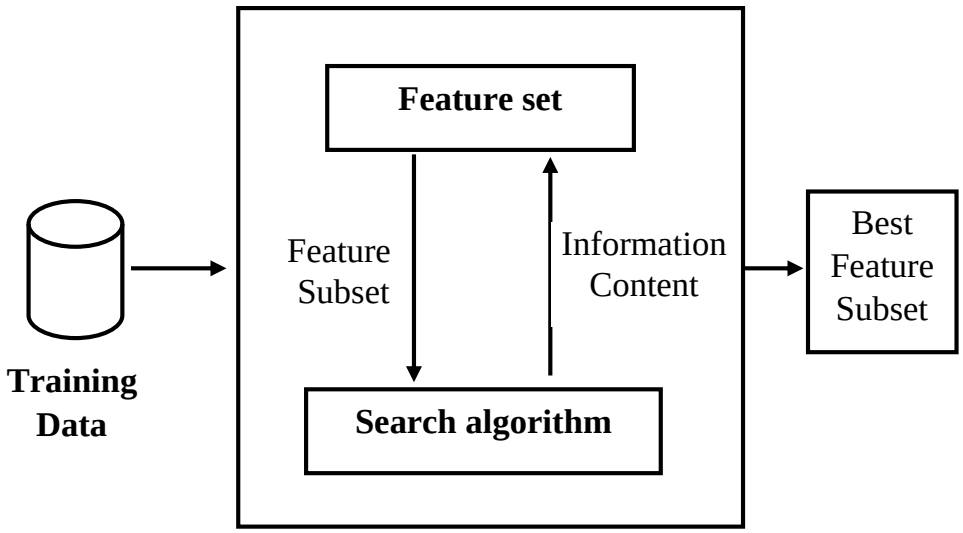


Figure 2.9. A general framework for filter methods

2.5.2. Wrapper Methods

However, wrapper methods utilize the predefined classifier to measure the quality of the currently selected subset of features. Here, the predefined classifier works as a black box. These methods are simple but powerful approaches to tackle the problem of feature selection (shown in Figure 2.10). However, they are usually time-consuming methods. So as to reduce computational time, some common feature search techniques are used like forward selection, backward elimination [73].

Forward selection begins with one feature, then at each iteration, it adds one feature if when adding this feature, it will improve performance of the classifier. This process repeats until all of features are checked. In backward elimination, it begins with all the features. In each iteration, it removes one feature if when removing this feature from feature set, it will enhance the performance of the classifier. This process repeats until there is no improvement in performance. Boruta package in R is one of the best example algorithms that applies the wrapper methods. Its algorithm is based a random forest classifier, and it uses Mean Decrease Accuracy to measure the important value for each feature. The feature with higher Mean Decrease Accuracy value is more important. At iteration, the feature with higher importance is checked firstly, then followed by features with lower important values.

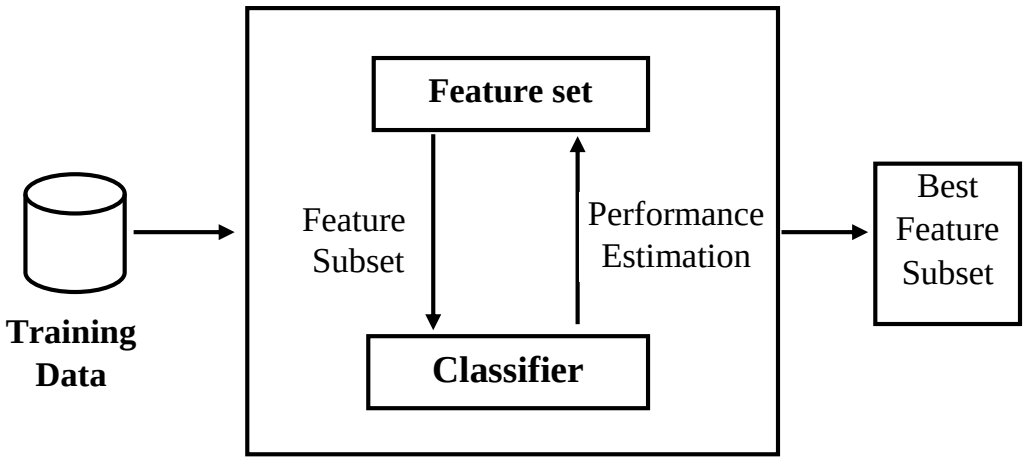


Figure 2.10. A general framework for wrapper methods

2.5.3. Embedded Methods

Embedded methods take the advantages of both wrapper methods and filter methods, their general framework shown in Figure 2.11.

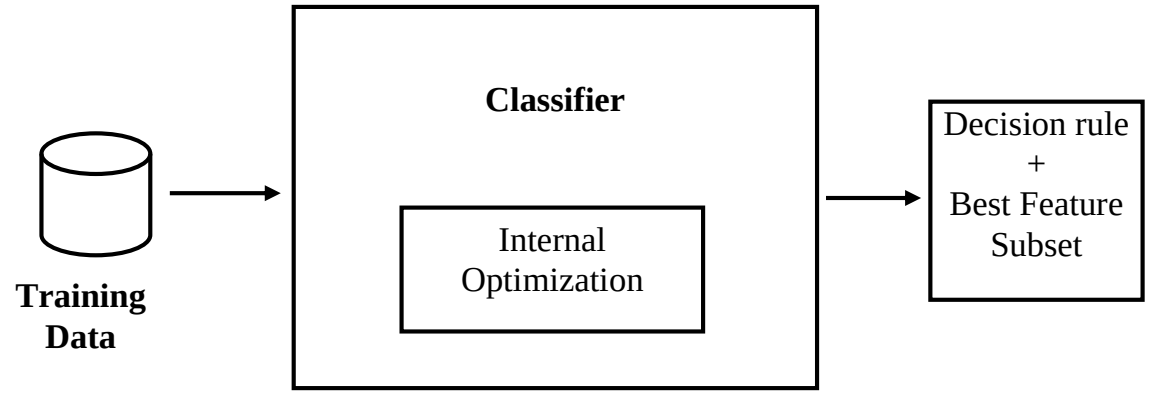


Figure 2.11. A general framework for embedded methods

2.6. Classification Evaluation

2.6.1. Classification Evaluation Metrics

Confusion matrix

In learning machine, the confusion matrix is a useful table that can help visualize the performance of a model (shown in Table 2.2). The six key terms are used in the confusion matrix and also used to calculate classification evaluation metrics.

Positives (P) is the number of positive samples. Negatives (N) is the number of negative samples. True positives (TP) is the number of the positive samples that were correctly classified by the classifier. True negatives (TN) is the number of the negative samples that were correctly classified by the classifier. False positives (FP) is the number of the negative samples that were incorrectly classified as positive. False negatives (FN) is the number of the positive samples that are misclassified as negative.

Table 2.2. Confusion matrix for a binary classifier

		Predicted class		
		Positive	**Negative**	Total
Actual class	**Positive**	TP	FN	P
	Negative	FP	TN	N
	Total	P'	N'	P + N

Evaluation Metrics

During classification training, to obtain the optimal classifier, evaluation metrics are so important, and a choice of right evaluation metrics for evaluating a classifier plays a crucial role as well [74]. Each evaluation metrics measures each characteristics of classification performance. Here, we describe some popular metrics used to evaluate a classifier.

Accuracy measures the ratio of samples which are correctly classified by the classifier.

$$Accuracy\ (Acc) = \frac{TP + TN}{P + N}$$

Error rate is also called misclassification rate that measures the percentage of samples that are misclassified by the classifier.

$$Error\ rate\ (Err) = \frac{FP + FN}{P + N}$$

Sensitivity is also named as true positive rate, recall (p) that measures the percentage of positive samples being classified as positive.

$$Sensitivity\ (Sn) = \frac{TP}{P}$$

Specificity is also named as true negative rate which evaluates the ratio of negative samples being classified as negative.

$$Specificity\ (Sp) = \frac{TN}{N}$$

Precision is used to measure the positive samples being classified as positive from the total classified samples in a positive label.

$$Precision\ (p) = \frac{TP}{TP + FP}$$

*F measure (*also called F_1, *F*-score) weighs a harmonic mean of precision and recall.

$$F_{measure} = \frac{2 * p * r}{p + r}$$

F_β *measure* is computed as a below equation, where β is a weight and non-negative real number.

$$F_\beta = \frac{(1 + \beta^2) * p * r}{\beta^2 * p * r}$$

Matthews correlation coefficient (MCC) takes the values in [-1, 1].

$$MCC = \frac{TP \times TN\ - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

2.6.2. Cross-Validation

k-fold cross-validation has been widely adopted to measure the performance of models. In this cross-valuation procedure, firstly, input data are randomly divided into k non-overlapping folds, Fd_1, Fd_2, …, Fd_k and each of them has nearly the same size. Then, we conduct learning and predicting k times. In the iteration i, the Fd_i is

used for predicting and the rest folds are used for learning [60]. Figure 2.12 is an example of 10-fold cross-validation procedure.

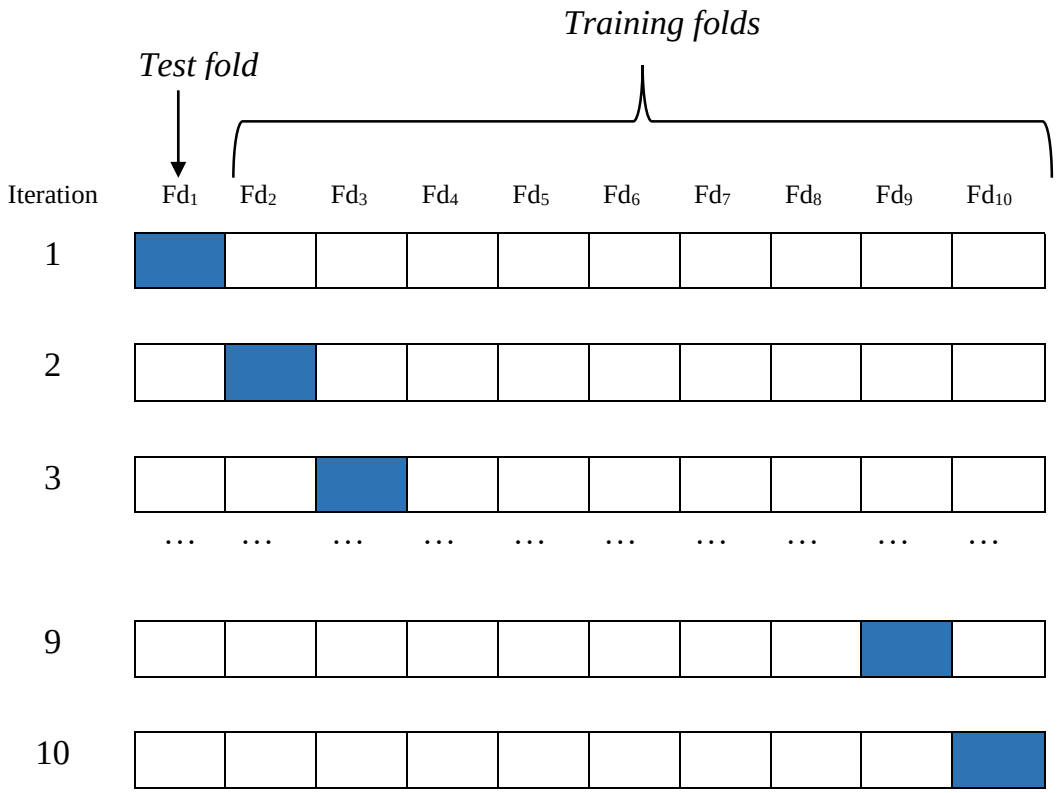


Figure 2.12. A procedure of 10-fold cross-validation.

The overall prediction accuracy can be calculated as a below equation.

$$Overall\ accuray = \frac{1}{10}\sum_{i=1}^{10} Acc_i$$

Where Acc_i is the prediction accuracy at the i^{th} iteration.

2.6.3. Leave-one-out cross-validation (LOOCV)

LOOCV is a specific type of k-fold cross-validation, in this case, k = n [60]. Figure 2.13 shows the procedure of leave-one-out cross-validation.

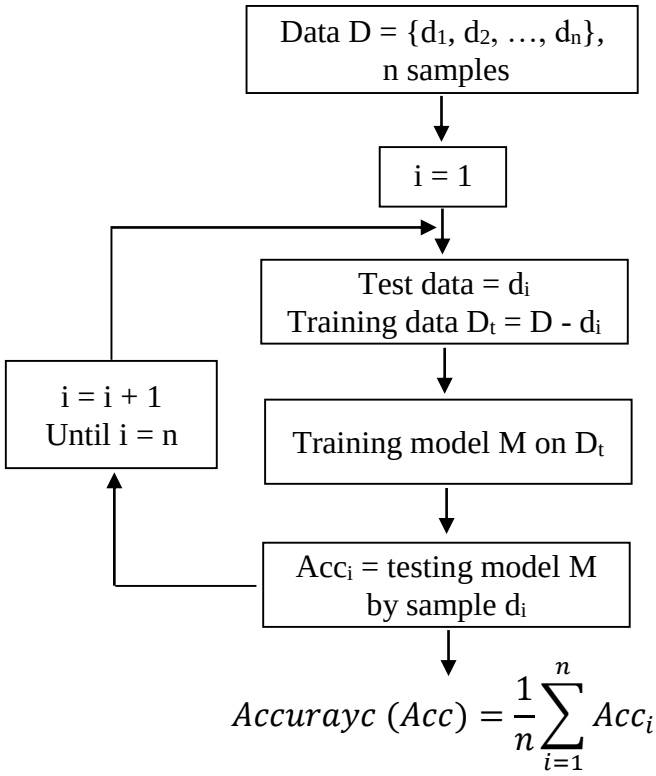


Figure 2.13. A procedure of leave-one out cross-validation

2.6.4. Bootstrap

The bootstrap is the procedure of sampling with replacement. There are different types of bootstrap method samplings, but one of the most well-known methods is 0.632 bootstrap method [60], [75] and works as follows. Suppose there is a dataset that has n samples, we sample the dataset n times with replacement. Then we produce another dataset of n samples called a bootstrap sampling or training dataset. Since some samples in the training dataset will repeat, there must be some original samples not contained in the training dataset. These samples are used to form a test dataset. If we try this out some times, consequently, on average, training dataset will include about 63.2% of original samples (the reason why it is called 0.632 bootstrap) and test dataset will include about 36.8% of original samples.

If we iterate the bootstrap sampling k times, in each iteration, we use a present training dataset to train the model and use a present test dataset to test model. The overall accuracy will be:

$$Acc(Model) = \frac{1}{k}\sum_{i=1}^{k}(0.632 \times Acc_i(Model)_{test\_data} + 0.368 \times Acc_i(Model)_{train\_data})$$

Where $Acc_i(Model)_{test\_data}$ is the prediction accuracy at the i[th] iteration when the model is evaluated on test dataset i. Where $Acc_i(Model)_{train\_data}$ is the prediction accuracy at the i[th] iteration when the model is evaluated on the training dataset i.

Chapter 3 : Materials and Methods

We start off by describing datasets used in our study in Section 3.1. These consist of splice dataset, promoter dataset and nucleosome positioning datasets. In section 3.2, we talk about the method converting biological sequences into numerical and categorical features. Our algorithm and a two-step feature selection technique are presented in the last two sections.

3.1. Datasets

To demonstrate the validity of our method in dealing with genetic sequence classification problem, we evaluated our approach on six datasets. The Table 3.1 shows the datasets in detail.

Table 3.1. Description of datasets

No	Dataset	Description	Number of Classes	Number of Sample	Sequence length (base)
1	Splice	Primate splice-junction sequences.	3	3175 (762+765+1648)	60
2	Promoter	*E. coli* promoter sequences	2	106 (53 + 53)	57
3	Human	*H. sapiens* nucleosomal and linker sequences	2	4573 (2273 + 2300)	147
4	Worm	*C. elegans* nucleosomal and linker sequences	2	5175 (2567 + 2608)	147
5	Fly	*D. melanogaster* nucleosomal and linker sequences	2	5750 (2900 + 2850)	147
6	Yeast	*S.cerevisiae* nucleosomal and linker sequences	2	3620 (1880 + 1740)	150

3.1.1. Promoter and Splice datasets

The two benchmark datasets from UCI, Splice and Promoter datasets, were firstly chosen for evaluation of our model. These datasets were also used in research [76]. The Splice dataset includes the splice-junction sequences. There are two types of splice junctions. The exon-intron “EI” is the part of DNA sequence ranging from the ending of an exon and the starting of an intron while intron-exon “IE” is a region of DNA between the ending of an intron and beginning of exon (as shown in Figure 3.1). The part of sequence which does not belong to “IE” and “EI” is called no junction “N”. This dataset is composed of 3,175 labeled samples and each sample has the length of 60 base pair.

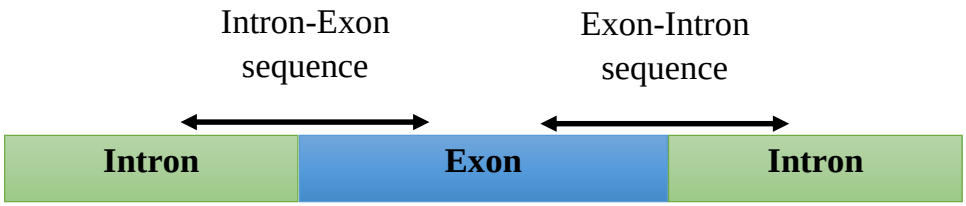


Figure 3.1. Splice-junction gene sequence

During RNA transcription process, transcription factors such as RNA polymerase and accessory proteins attach to the promoter, then carry out the initiation of transcription. Promoter parts are DNA sequences located adjacent to the initial sites of transcription. Promoter dataset consists of 106 labeled promoter sequences (positive and negative), with length of 57 base pair. Positive samples are promoter sequences whereas negative sequences are non-promoter sequences (see Figure 3.2).

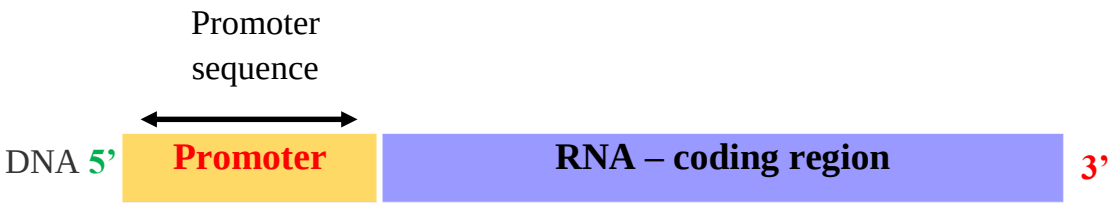


Figure 3.2. Promoter sequence in DNA sequence.

3.1.2. Nucleosome benchmark datasets

The other four datasets are about nucleosome forming and inhibiting sequences in four species: Human, worm, fly and yeast. The first three datasets were collected by Guo et al. [13]. They were downloaded from their website. These datasets were previously used in the research [13], [53], [54]. Positive samples are nucleosome-forming sequences. Negative samples are nucleosome-inhibiting sequences. Human, worm and fly consist of 4573, 5175 and 5750 samples, respectively. The number of positive and negative samples is shown in Table 3.1. All sequences in these three datasets have the same length of 147 base pair. In addition, Yeast (Saccharomyces cerevisiae) dataset consists of 1880 positive samples and 1740 negative samples. Each of these sequences has the length of 150 base pair, and this dataset was also used in [54], [77], [78].

Nucleosome forming sequence (nucleosome DNA) and inhibiting sequence (linker DNA) in a chromatin are shown in Figure 3.3.

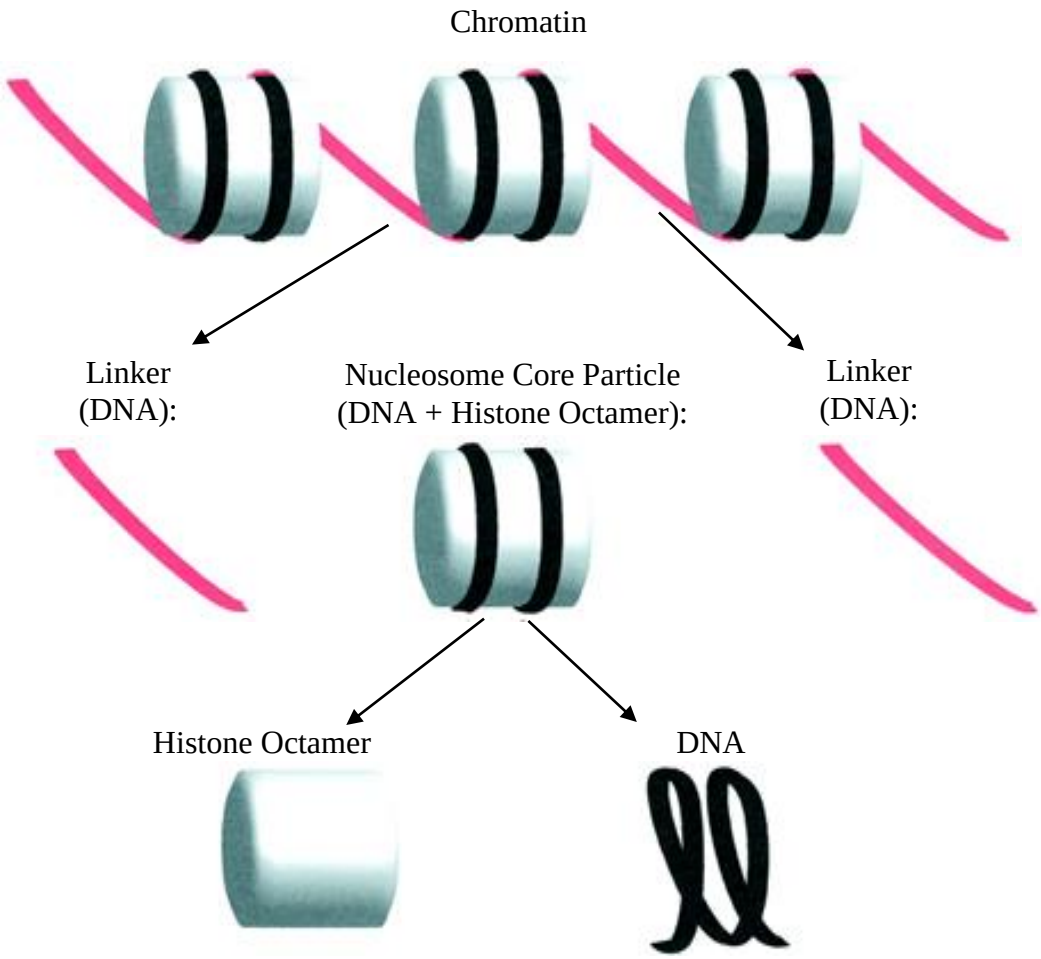


Figure 3.3. The overview of the basic entities forming chromatin. (Source [79])

3.2. Features

Data transformation is defined as a process that converts data from the original format to the target format. In order to apply the proposed model, the DNA sequence datasets need to be transformed into required formats. The Block A in Figure 3.4 converts each DNA sequence from DNA datasets into a combination vector.

In this research, we used the combination of the five different vectors named as 1-categorical vector (1CAT), 2-categorical vector (2CAT), 2-mer vector (2MER), 3-mer vector (3MER), and 4-mer vector (4MER). Given a biological sequence s of length n, $S_1S_2 \ldots S_n$, where $S_i \in \{A, C, G, T\}$. These vectors can be defined as follows:

1-categorical vector (1CAT)

1CAT = (A_1, A_2, …, A_n), where A_i is a nucleotide at i^{th} position, i = 1, 2, …, n. For example, s = AGGTCCTACT, 1CAT will be:

A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
A	G	G	T	C	C	T	A	C	T

2-categorical vector (2CAT)

2CAT = (B_1, B_2, …, B_{n-1}), where B_i is two consecutive nucleotides at i^{th} and $(i+1)^{th}$ positions, i = 1, 2, …, n-1. Following is an example of 2CAT vector for above s sequence.

B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9
AG	GG	GT	TC	CC	CT	TA	AC	CT

2-mer vector (2MER), 3-mer vector (3MER), and 4-mer vector (4MER)

In terms of biological sequence, k-mers can be defined as all possible subsequences of length k within a sequence. A k-mer is a string of k successive nucleotides contained the genetic sequence and there are 4^k possible k-mers: s_1, s_2, …, s_{4^k}. The k-mer vector denoted as kMER is defined by kMER = ($c[s_1]$, $c[s_2]$, …, $c[s_{4^k}]$), where $c[s_i]$ is a number of times that s_i occurs in s, i = 1, 2, …, 4^k. Therefore, using sequence s above, 2MER will be:

AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
0	1	1	0	0	1	0	2	0	0	1	1	1	1	0	0

and 3MER will be:

AAA	AAC	AAG	AAT	ACA	ACC	ACG	ACT	AGA
0	0	0	0	0	0	0	1	0

AGC	AGG	AGT	ATA	ATC	ATG	ATT	CAA	CAC
0	1	0	0	0	0	0	0	0

CAG	CAT	CCA	CCC	CCG	CCT	CGA	CGC	CGG
0	0	0	0	0	1	0	0	0

CGT	CTA	CTC	CTG	CTT	GAA	GAC	GAG	GAT
0	1	0	0	0	0	0	0	0

GCA	GCC	GCG	GCT	GGA	GGC	GGG	GGT	GTA
0	0	0	0	0	0	0	1	0

GTC	GTG	GTT	TAA	TAC	TAG	TAT	TCA	TCC
1	0	0	0	1	0	0	0	1

TCG	TCT	TGA	TGC	TGG	TGT	TTA	TTC	TTG	TTT
0	0	0	0	0	0	0	0	0	0

3.3. Algorithm

The proposed algorithm consists of four main steps. The flowchart of our algorithm is shown in Figure 3.4, and works as below:

1) Block A in Figure 3.4 is in charge of converting DNA sequences into feature vectors. Each DNA sequence is encoded into a combination vector that contains five different groups of features: 1CAT, 2CAT, 2MER, 3MER, and 4MER.

2) At Block B in Figure 3.4, feature ranking is conducted by the randomForest function for R [80].

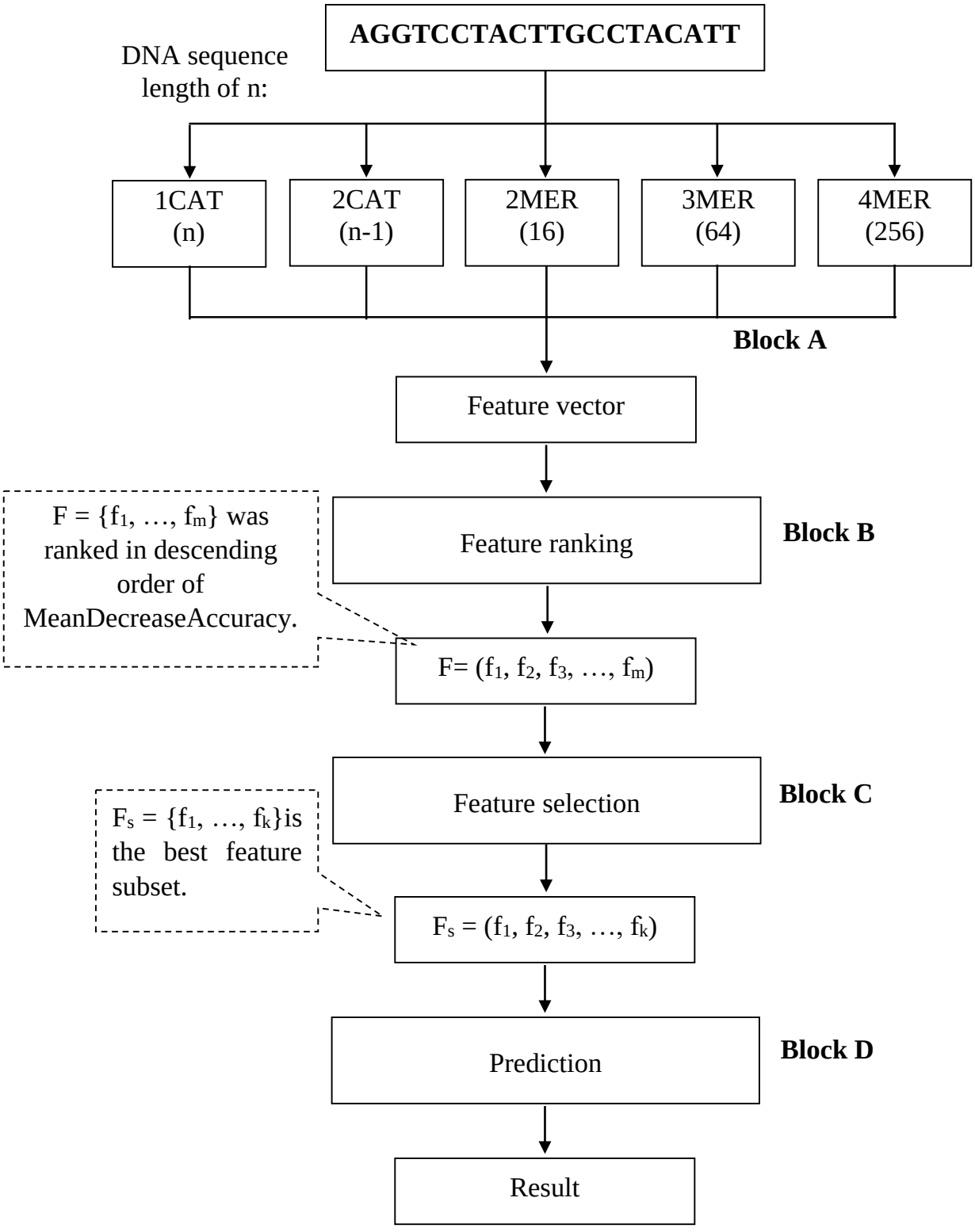


Figure 3.4. The flowchart of the proposed algorithm.

3) Block C in Figure 3.4 is responsible for feature selection by performing learning and predicting with the ksvm function for R in kernlab package [81]. Each feature subset is evaluated by the average of prediction accuracies of 10-fold cross-validation.

4) Block D in Figure 3.4 is in charge of predicting by using the best feature subset $\{f_1, \ldots, f_k\}$ obtained in the previous step. We also evaluate the model by 10-fold cross-validation ten times. Herein, the best feature subset is the feature subset with the best accuracy.

3.4. Feature Selection

Nowadays, there has been a remarkable increase in the number of researches exploiting feature selection techniques. They have been used in supervised learning as well as unsupervised learning. Their aims are threefold. The first goal is to avoid overfitting and enhance performance. The second advantage is to reduce computational time and space required to execute models, and the final goal is to identify which features are relevant to a problem and to gain a deeper insight into the data.

The feature selection approach used in our research is a kind of greedy algorithm, and works as two following steps:

Step 1: With pre-calculated feature set $F = \{f_1, f_2, \ldots, f_m\}$ being ranked in descending order of MeanDecreaseAccuracy, we evaluate a feature subset $\{f_1, f_2, \ldots, f_{10}\}$, a feature subset $\{f_1, f_2, \ldots, f_{20}\}$, and so forth, until a feature subset $\{f_1, f_2, \ldots, f_m\}$ by conducting training and predicting with ksvm function [81] (shown in Figure 3.5).

Step 2: Neighbors of the feature subset with the best accuracy of prediction in the preceding step are tested.

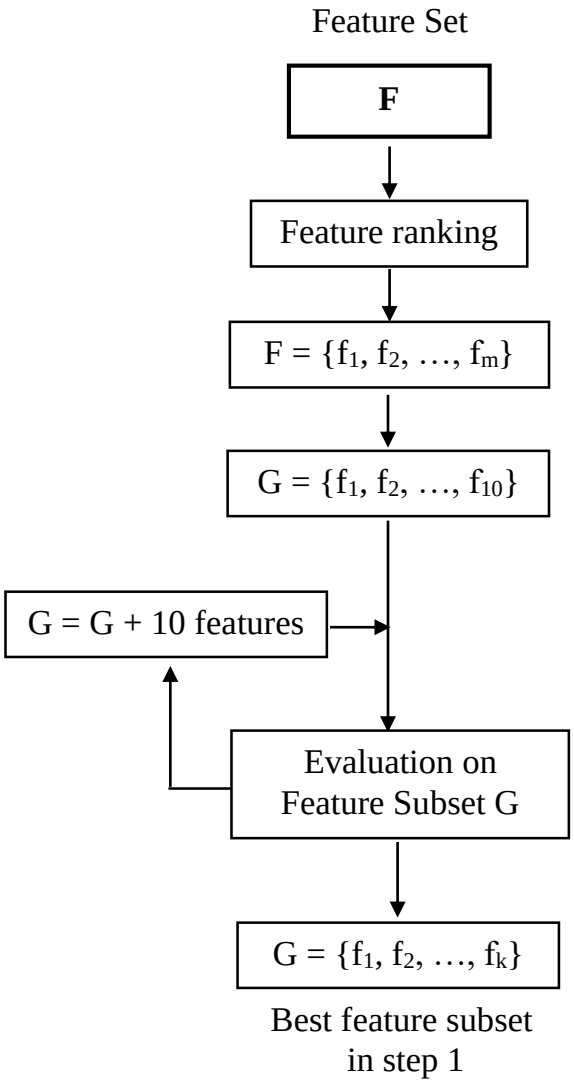


Figure 3.5. Step 1 of the feature selection algorithm.

Chapter 4 : Experimental Results and Discussion

Firstly, feature ranking and list top 10 of important features in each dataset are shown in the section 4.1. Then, we present the prediction accuracies in the first step of the two-step algorithm in section 4.2. The prediction accuracies in the second step of the two-step algorithm are demonstrated in section 4.3. Four evaluation metrics and evaluation methods used in our research are introduced in section 4.4. Next, we summarize the state-of-the-art models which were used to compare with our model in section 4.5. We also perform the comparison of our results with results of previous model in this section. Finally, discussion and conclusion are given in the section 4.6.

4.1. Feature Ranking by Random Forest

Random forests are well-known ensemble learning method which can conduct both classification and regression. Apart from these tasks, the randomForest function for R in randomForest package [80] can measure the importance of all features by MeanDecreaseAccuracy or MeanDecreaseGini values. In this research we adopted the MeanDecreaseAccuracy value as the importance of features. The relationship between rank and MeanDecreaseAccuracy normalized into the range of [0, 1] in each dataset is shown in the Figure 4.1.

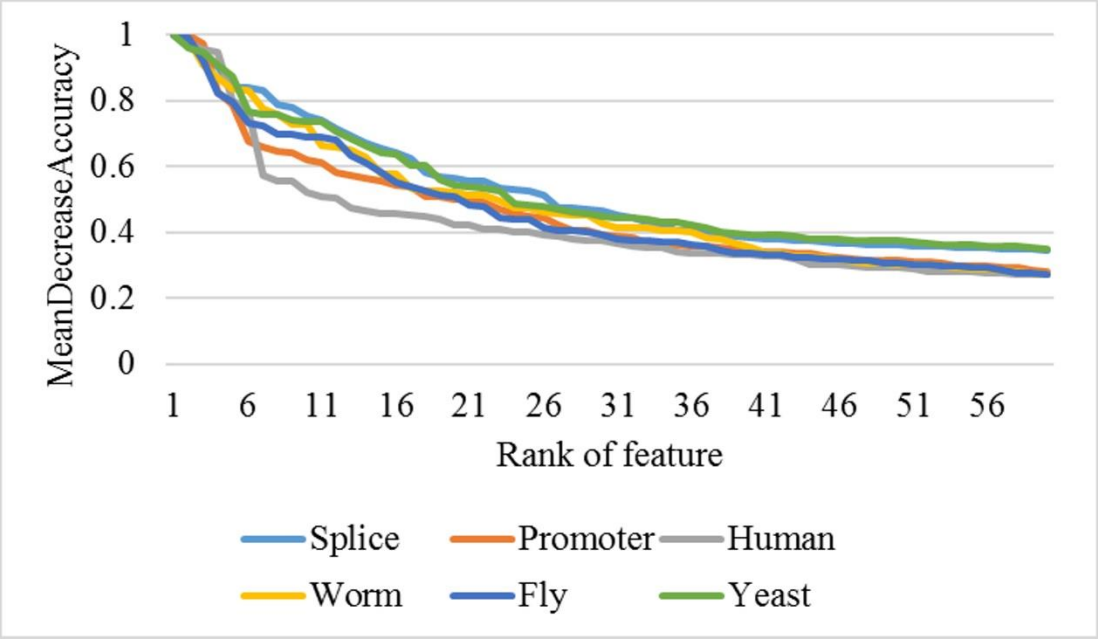


Figure 4.1. MeanDecreaseAccuracy along feature ranking from top 1~ 60.

In general, there is a sharp decrease in the importance of features in Promoter and human datasets in the areas of top 1~5. This is then followed by a steady decline trend in the rest region. With Splice, worm and fly datasets, the importance of features fall slowly in the region of from top 1 to 16. The importance in the remainder declines gradually.

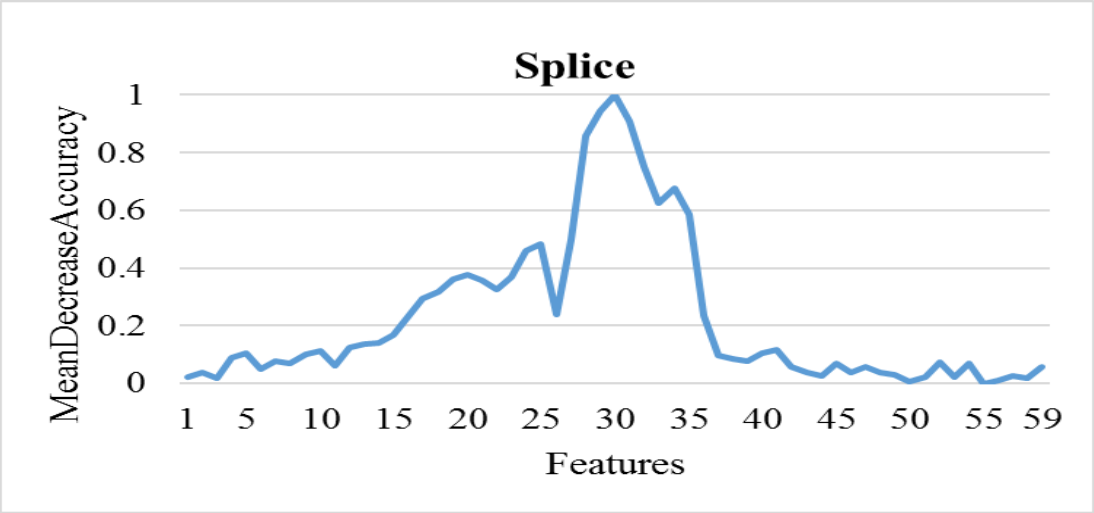
Features with high importance in validation datasets are listed in Table 4.1. From this table it is clear that human, worm, and fly datasets have features with high importance containing mainly “A” (adenine) and “T” (thymine). However, the percentage of “C” (cytosine) and “G” (guanine) increases slightly in the fly dataset. More observations are that TTT, TTTT, AA, AAA, AAAA, AAAT, ATTT

features are highly important in human and worm datasets. It is similar to the case of fly dataset where TTT, TT, TTTT, ATA, AAAA, TAT are so important. AAAA, TTTT, TA, AAA, TTT, TAT, ATA are highly important for yeast dataset. This coincides with the results in the research of Higashihara *et al*. [10] for classification of nucleosome datasets. The research showed that T and A were both highly important. Additionally, in Table 4.1, it was partially demonstrated that the combination of numerical and categorical might be effective. In case of worm dataset, the first and the fourth important features are categorical (B_1 and A_1), and others are numerical.

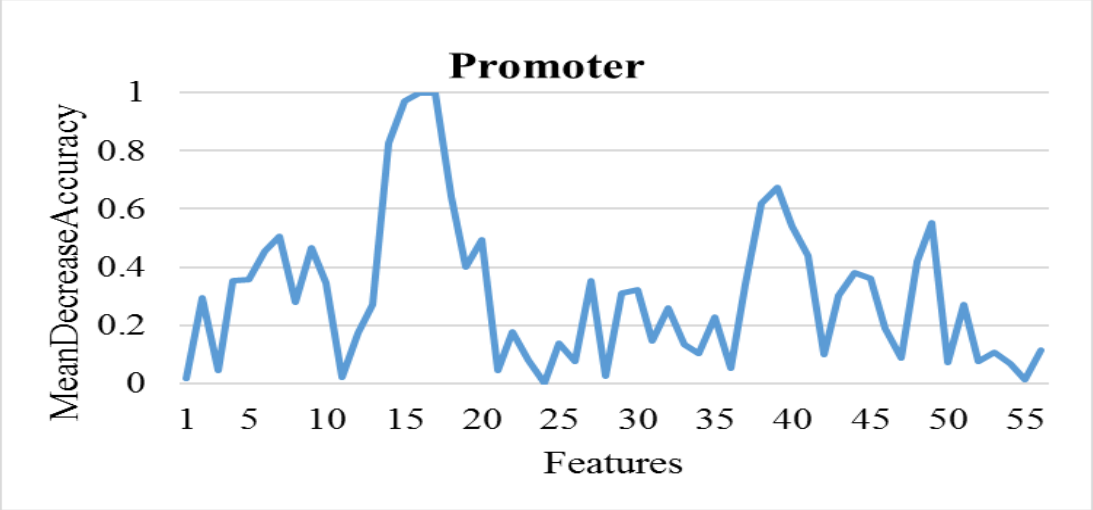
Table 4.1. List of important features.

No	Dataset	List of top 10 features with high importance sorted by descending order of rank
1	Splice	B_{30}, B_{29}, B_{31}, B_{28}, A_{29}, A_{30}, B_{32}, A_{32}, B_{34}, A_{31}
2	Promoter	B_{17}, B_{16}, B_{15}, B_{14}, A_{15}, B_{39}, A_{17}, B_{18}, A_{16}, B_{38}
3	Human	TTTT, AAA, TTT, AAAA, TT, AA, AAAT, ATTT, TG, TAAA
4	Worm	B_1, AAA, AA, A_1, TTT, AAAA, AAAT, TTTT, ATTT, AATT
5	Fly	TA, GC, CG, TTT, TT, TTTT, ATA, CA, AAAA, TAT
6	yeast	AAAA, TTTT, TA, AAA, TTT, TAT, ATA, CGCG, CA, TT

For Splice and Promoter datasets, however, features in 2CAT vectors are so important. B_{30}, B_{29}, B_{31}, B_{28} features are highly important in Splice dataset, which means that the nucleotides around the center of splice sequences play a vital role in prediction. This finding agrees with the structure of splice site sequences where splice-junctions are at the middle of sequences. B_{17}, B_{16}, B_{15}, B_{14} are highly important in Promoter dataset. Figure 4.2 demonstrates the relationship between features in 2CAT vectors of Splice and Promoter datasets and MeanDecreaseAccuracy normalized into the interval of [0, 1]. The figure illustrates that the highly important features in 2CAT vector of Splice dataset are located at the region of from 25 to 35. While those of Promoter dataset settled at the area of between 12 to 18.



(a)



(b)

Figure 4.2. MeanDecreaseAccuracy of features in 2CAT vector on (a) Splice and (b) Promoter datasets.

4.2. Prediction Accuracy of Feature Subsets along Ranking

As described in step 1 in section 3.4 of Chapter 3, feature subsets along the ranking were assessed by support vector machine. With human, worm and fly datasets, there are 63 different feature subsets at intervals of 10: $\{f_1, f_2, \ldots, f_{10}\}$, $\{f_1, f_2, \ldots, f_{20}\}$, $\{f_1, f_2, \ldots, f_{30}\}$, …, $\{f_1, f_2, \ldots, f_m\}$ being tested. The prediction accuracy is based on the average accuracy of 10-fold cross-validation. However, there are around 45 feature subsets for Promoter dataset and Splice dataset. Table 4.2. demonstrates the results of prediction.

Table 4.2. Prediction accuracies obtained by using either the whole set of features and the best feature subset in step 1.

No	Dataset	The whole set of features		The best feature subset in step 1		Improvement (%)
		# Feature	Acc (%)	# Feature	Acc (%)	
1	Splice	455	94.55	40	96.77	2.22
2	Promoter	449	94.34	90	100	5.66
3	Human	629	85.94	420	86.35	0.41
4	Worm	629	89.06	180	89.28	0.22
5	Fly	629	80.16	140	81.79	1.63
6	Yeast	635	100	30	100	0.00

4.3. Prediction accuracy of neighbors around the best feature subset

With best feature subset obtained at step 1 in section 3.4 of Chapter 3, we conducted step 2. Table 4.3 illustrates results and the number of features in this step.

Table 4.3. Prediction accuracies in step 2 compared with those in step 1.

No	Dataset	The best feature subset in step 1		The best feature subset in step 2		Improvement (%)
		# Feature	Acc(%)	# Feature	Accuracy (%)	
1	Splice	40	96.77	48	96.93	0.16
2	Promoter	90	100	90	100	0
3	Human	420	86.35	428	86.49	0.14
4	Worm	180	89.28	177	89.53	0.25
5	Fly	140	81.79	148	81.93	0.14
6	Yeast	30	100	22	100	0.00

Figure 4.3 illustrates the proportion of feature groups in the best feature subsets of four datasets: Splice, Promoter, human and worm.

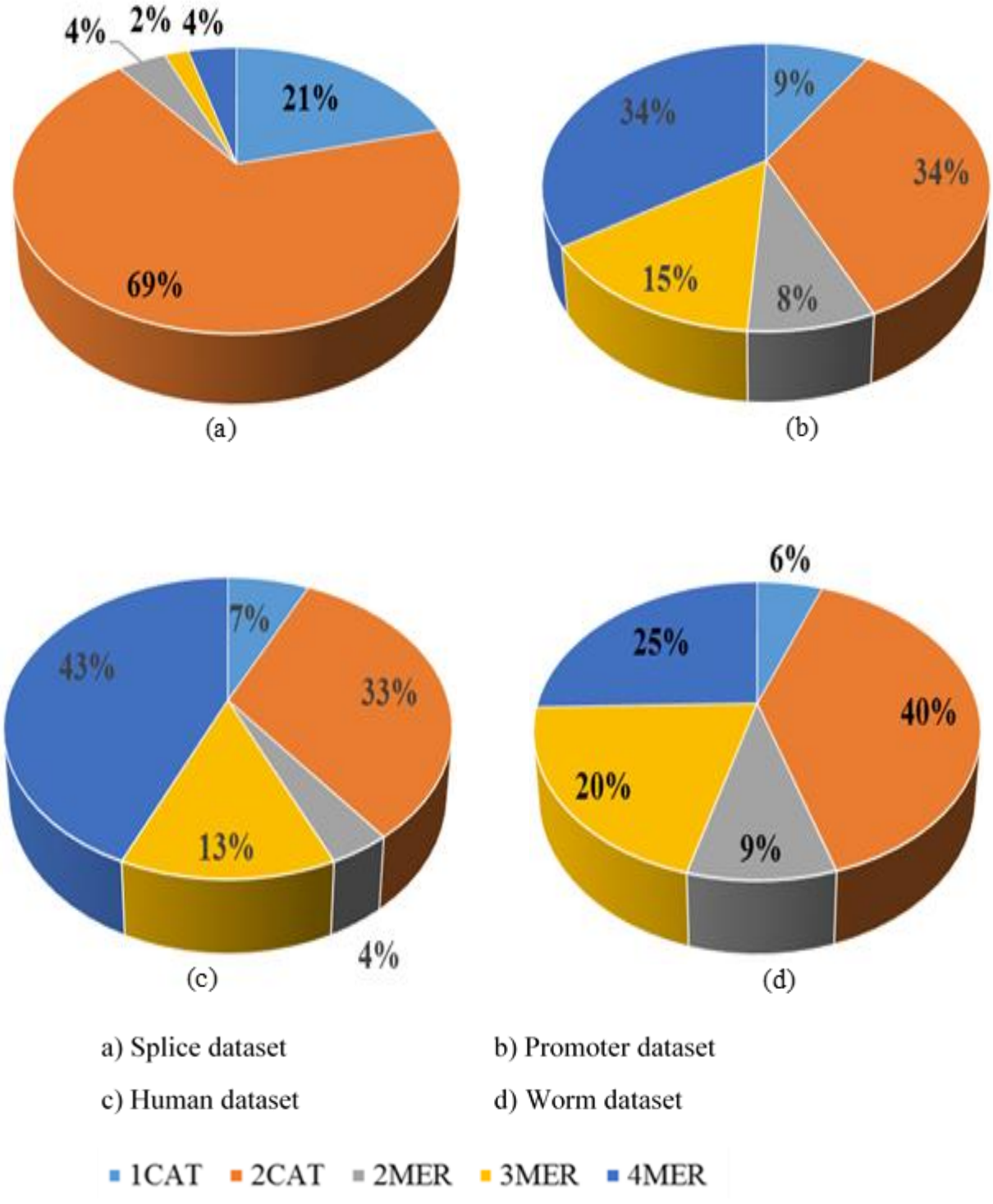


Figure 4.3. The percentage of feature groups in the best feature subsets.

4.4. Evaluation

4.4.1. Evaluation Metrics

To evaluate the quality of our method, four following metrics were used.

$$Accuracy\ (Acc) = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Sensitivity\ (Sen) = \frac{TP}{TP + FN}$$

$$Specificity\ (Sp) = \frac{TN}{TN + FP}$$

$$Matthews\ correlation\ coefficient\ (MCC) = \frac{TP \times TN\ - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

4.4.2. Performance Evaluation of the Method

Using the best feature subsets achieved at the step 2 of the two-step feature selection approach (in section 3.4, Chapter 3), we applied our model to classify the DNA sequences in the validation datasets and compared its performance with the previous researches. For evaluation, we mainly carried out 10-fold cross-validation ten times, and then computed average prediction results. With Promoter data, however, we employed leave one out ten times due to the fact that the number of its samples is small, 106 samples.

4.5. Comparison with other methods

4.5.1. Summary of existing models

For Splice and Promoter datasets, we made the comparison our model with the previous model conducted by Nguyen *et al.* [76]. The results from this research are known as the best performance prior to our research. The motivation behind this model was the desire to apply a deep learning model for text classification to DNA sequence classification. At first, the researchers translated DNA sequence into sequence of words as a text sentence, then applied the representation technique for text to this produced sequence. Lastly, two-dimensional matrices representing

DNA sequences using one hot vectors were directly used as input to the CNN algorithm (see Figure 4.4).

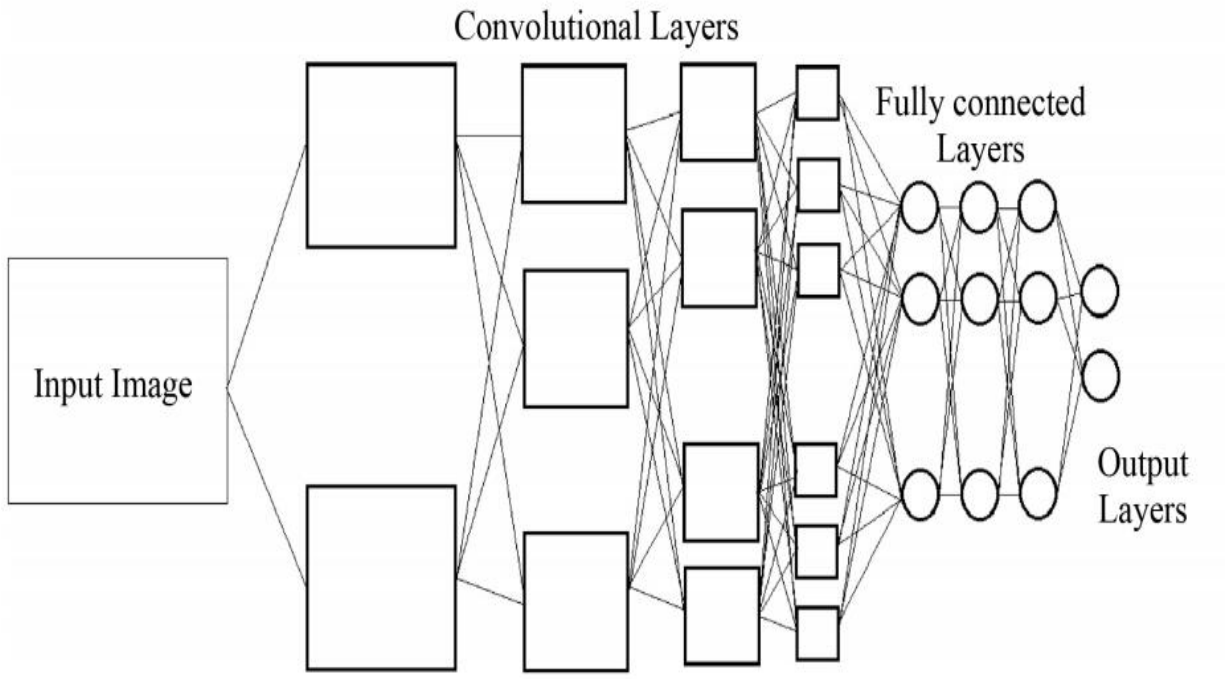


Figure 4.4. The structure of convolutional neural network. (source *[76]*)

However, for the first three nucleosome positioning datasets (human, worm, fly), we compared our results with the results taken from researches [13], [53], [54]. These models are summarized as below:

– iNuc-PseKNC predictor was proposed by Guo *et al.* in 2014 [13]. It included of following steps.
  - DNA sequences, firstly, were encoded into pseudo k-tuple nucleosome composition features.
  - Then, support vector machine was applied to classify DNA nucleosome sequences.

– iNuc-PseSTNC in 2016 [53] was introduced by Tahir and Hayat:
  - DNA sequences were converted into three types of features: 2-mer composition, 3-mer composition and split 3-mer composition.
  - Next, support vector machine was applied

– 3LS and TNS models were presented by Awazu [54]:

- DNA sequences are converted into: Three different groups of numerical features.
- Based on the linear regression model, the author developed the models to classify DNA nucleosome sequences.

For yeast dataset, we compared our results with those taken from [54], [77], [78]. In 2015, Chen *et al.* [77] developed the model by using DNA deformation energy. Yi *et al.* [78] in 2012 introduced the model which applied the nearest neighbor algorithm.

4.5.2. Comparison on Promoter and Splice datasets

Table 4.4 shows that there were significant improvements in the prediction accuracy of our method for both datasets. In particular, the prediction accuracy of the proposed model increased by 0.94% and reached the peak of 100% on Promoter dataset. This means that all samples in this dataset were correctly predicted by our proposed model. This result has not obtained by any previous methods. Our method also achieved the high prediction accuracy for Splice dataset with 96.81%.

Table 4.4. Accuracies of the proposed model compared to those in *[76]*.

No	Dataset	Acc (%) in [76]			Acc (%) by our method			Improvement in average (%)
		Min	Max	Average	Min	Max	Average	
1	Splice	95.87	96.73	96.18	96.65	96.93	96.81	0.63
2	Promoter	99.06	99.06	99.06	100	100	100	0.94

For Promoter dataset, we performed a further comprehensive comparison with other reported studies in terms of TP, FN, FP, TN, Acc and (MCC). Table 4.5 illustrates results in detail.

Table 4.5. Comparison our method with other reported methods.

Prediction method	**Reference**	**TP**	**FN**	**FP**	**TN**	**Acc(%)**	**MCC**
Our method	This paper	**53**	**0**	**0**	**53**	**100**	**1.00**
Expectation Maximization and Support Vector Machine(EMSVM)	Maleki *et al.* [45]	52	1	0	53	99.05	0.98
Hidden Markov Model(HMM)	Tavares *et al.* [82]	50	3	5	48	92.45	0.850
Complement Class Naive Bayes (CNB)		49	4	3	50	93.40	0.868
Multilayer Perceptron Neural Network (MLP)		49	4	3	50	93.40	0.868
Support Vector Machine(SVM)		49	4	3	50	92.45	0.849
LogitBoost		47	6	5	48	89.62	0.793
NBTree		47	6	5	48	89.62	0.793
Lazy Bayesian Rules Classifier(LBR)		48	5	3	50	92.45	0.850
PART		44	9	11	42	81.13	0.623

Figure 4.5 shows ROC space including our method and reported methods. In the ROC space, the x axe is FP rate, 1- specificity. The y axe is the TP rate, sensitivity. The point (0, 1) in a ROC space shows the excellent classification algorithm, which means that the classification reaches the sensitivity of 100% and specificity of 100%. Our method is represented by the point (0, 1) that is the best point in ROC space compared with previous methods.

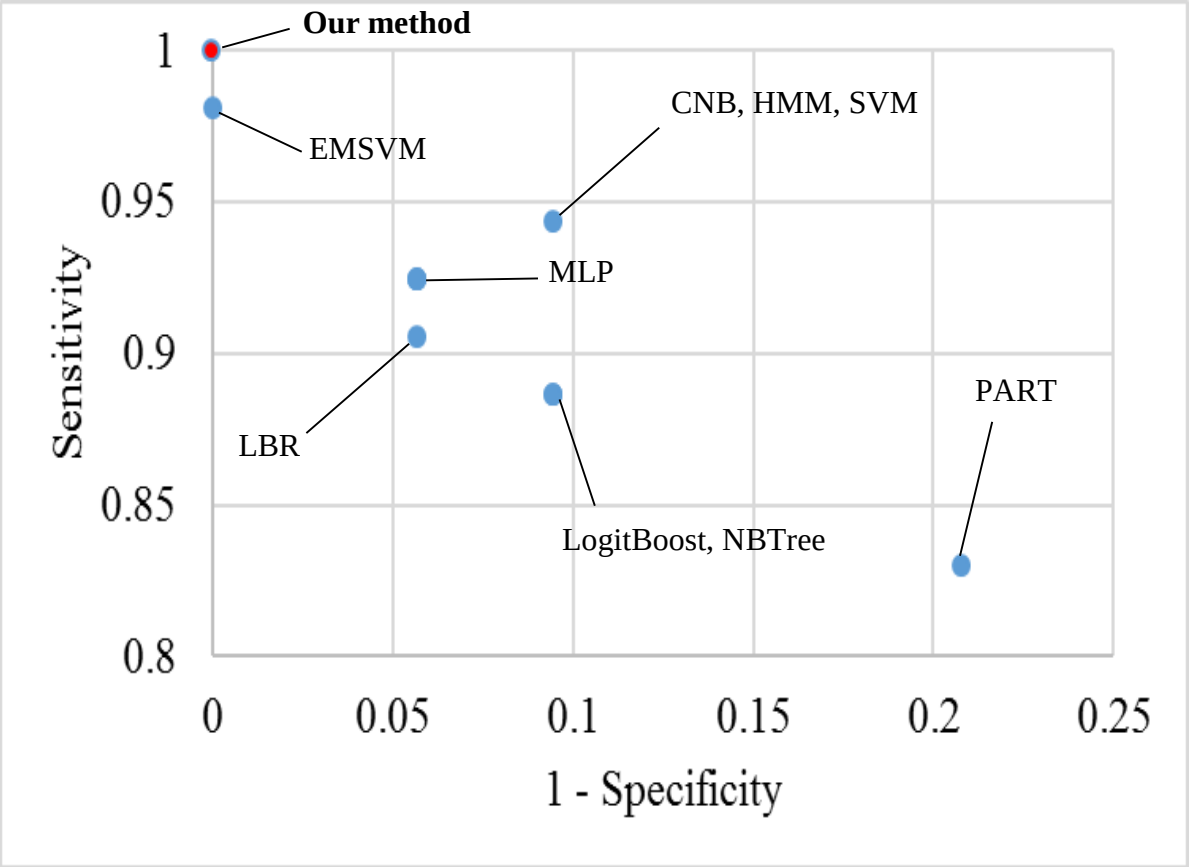


Figure 4.5. Our method and reported methods in ROC space.

4.5.3. Comparison on nucleosome positioning datasets

With human, worm and fly datasets, we compared our models to methods in [13], [53], [54] on four metrics: Accuracy, sensitivity, specificity and Matthews correlation coefficient. Table 4.6 indicates the results of all methods in detail.

From this table, the first noticeable thing is that for yeast dataset, our method and TNS completely outperformed the previous methods. Our model achieved the Acc of 100%, Sen of 100%, Sp of 100% and MCC of 1.0. The second result is worth pointing out that our method outperformed all of competing methods on worm dataset with Acc of 89.35%, Sen of 92.45% and MCC of 0.79. The third thing to note is that on the fly dataset our model also achieved better results than those of the other previous models with Acc of 81.75%, Sen of 79.14%, Sp of 84.40% and MCC of 0.64 except 3LS. Moreover, on human dataset, the prediction Acc of the proposed method (86.33%) was higher than that of iNuc-PseKNC, TNS but lower than iNuc-PseSTNC and 3LS.

Table 4.6. Performance comparison of our model and previous models.

Dataset	Method	Acc (%)	Sen (%)	Sp(%)	MCC
Human	Our method	86.33	89.77	82.93	0.73
	iNuc-PseKNC [13]	86.27	87.86	84.70	0.73
	iNuc-PseSTNC [53]	87.60	89.31	85.91	0.75
	3LS [54]	**90.01**	**91.69**	**88.35**	**0.80**
	TNS [54]	81.67	-	-	-
Worm	Our method	**89.35**	**92.45**	86.30	**0.79**
	iNuc-PseKNC [13]	86.90	90.30	83.55	0.74
	iNuc-PseSTNC [53]	88.62	91.62	86.66	0.77
	3LS [54]	87.86	86.54	**89.21**	0.76
	TNS [54]	83.94	-	-	-
Fly	Our method	81.75	79.14	**84.40**	0.64
	iNuc-PseKNC [13]	79.97	78.31	81.65	0.60
	iNuc-PseSTNC [53]	81.67	79.76	83.61	0.63
	3LS [54]	**83.41**	**84.07**	82.74	**0.67**
	TNS [54]	70.82	-	-	-
Yeast	Our method	**100**	**100**	**100**	**1.00**
	TNS [54]	**100**	-	-	-
	Chen et al. [77]	98.10	98.20	98.00	0.96
	Yi et al. [78]	99.06	-	-	-

Although iNuc-PseKNC model achieved the same MCC (0.73) with our model, Acc and Sen of our method were better than those of iNuc-PseKNC except for Sp.

As a whole, in the terms of accuracy and Mathews correlation coefficient, our method performed better than all of active methods on yeast and worm datasets.

4.6. Discussion and Conclusion

The combination vector can reflect not only the positional information (categorical features) of DNA sequence, but also the quantitative information (k-mer features) of sequence. It can characterize a genetic sequence. Moreover, we utilized the ability of executing categorical data and numerical data of random forests and SVM to solve our problem. We also made use of the advantages of random forest in automatically producing variable importance to rank features, then applied the feature ranking to conduct feature selection. The used feature selection technique is a greedy technique which does not learning and predicting on all possible feature subsets. This can reduce dramatically computational cost. However, one limitation of this model is that all DNA sequences in one dataset need to be the same length.

In this research, we proposed a simple but powerful model for solving DNA sequence classification problems. The model was tested on six different datasets: Splice, Promoter, human, worm, fly, and yeast datasets. On Splice and Promoter datasets, the experimental results show that there was a significant increase in the performance of our model. Especially, the proposed model reached the accuracy of 100% on Promoter and yeast datasets.

We also compared our model with the other four models: iNuc-PseKNC [13], iNuc-PseSTNC [14], TNS and 3LS [15]. In terms of accuracy, sensitivity and MCC, our method achieved better performance than any other competing method for predicting nucleosome positioning in worm genome. For fly genome, the proposed method also outperformed the other methods except 3LS model. For predicting nucleosome positioning in human genome, our method performance was

higher than iNuc-PseKNC and TNS, but lower than the other two models. Therefore, it can be concluded that our model is effective for fixed-length DNA sequence classification.

Chapter 5 : Summary and Future Research

The research context, objectives, materials, methods and experimental results of the proposed model were introduced and shown in the previous chapters. In this chapter, we would like to summarize this thesis and propose some directions we will fulfil in the future.

5.1. Dissertation summary

Research context, objectives and contributions

In the past decades, there has been an ever-increasing number of methods for classification of DNA sequences. There have been researches that used numerical features or categorical features for classifying DNA sequences, however, until now numerical and categorical features were separately studied. Therefore, the combination of these two types of features was considered in our thesis. The target of our research is to combine five feature vectors (such as 1-categorical vector, 1-categorical vector, 2-mer vector, 3-mer vector, and 4-mer vector) to address the problem for classifying fixed-length DNA sequences. The specific objectives and contributions of present thesis are:

- To develop an effective framework for classification of DNA sequence on three types of datasets: splice site, promoter and nucleosome positioning datasets.
- To enhance performance by conducting the greedy feature selection algorithm.
- To find which group of features are more effective in each dataset.

Materials and Methods

To obtain the above stated objectives, this thesis developed a model to classify DNA sequences by combining five groups of features, two of them are categorical features and the other three are numerical features. So as to achieve better the performance, the two-step feature selection algorithm was also utilized.

The proposed model in present thesis was evaluated on six benchmark datasets. The first one is Splice dataset being about primate splice-junction sequences. The second dataset is promoter dataset being about *E. coli* promoter sequences. Other four datasets are nucleosome positioning datasets of four species (human, worm, fly and yeast) containing nucleosome forming and inhibiting sequences.

Experimental Results

By conducting feature selection on the mixture of five feature vectors, we could also find which type of features are more effective in each dataset. The findings

of this study are consistent with the previous research. Moreover, the results also showed that some parts of DNA sequences are so important for improving the accuracies on some datasets.

Four evaluation metrics (accuracy, sensitivity, specificity and Mathews correlation coefficient), 10-fold cross-validation were used to weigh our model. Through the performance evaluation on six benchmark datasets of fixed-length DNA sequences, our algorithm achieved comparable or higher performance than other advanced algorithms. The most thing to note is that our model reaches the accuracy of 100 % on two datasets, promoter and yeast.

5.2. Future Research

In this thesis, we proposed the simple but powerful framework for classification of fixed-length DNA. Therefore, we are going to apply this model to other areas of sequence recognition like protein classification or combine categorical features used in the present thesis with other numerical feature vectors to improve the performance of fixed-length DNA sequence classification.

Application of the proposed model to protein prediction.

For predicting beta-turns and beta-turn types, the combination of categorical features with the below numerical features will be considered. These features are Position Specific Scoring Matrices (PSSMs), predicted shape strings, and predicted protein blocks.

For phosphorylation site prediction, we will combine categorical features with other numerical features used in the research of Ismail *et al.* [83].

Improving the performance of DNA sequence classification.

We will incorporate other numerical features used by previous studies into our model. These features consist of:

PseKNC (stand for "Pseudo k-tuple nucleotide composition") [13]

Given a biological sequence $S = R_1R_2R_3 \ldots R_n$ length of n, $R_i \in \{A, C, G, T\}$.

PseKNC features of S is:

$$D = \{d_1, d_2, \dots, d_{4^k}, d_{4^k+1}, \dots, d_{4^k+\lambda}\}$$

where d_u is calculated as following:

$$d_u = \begin{cases} \dfrac{f_u}{\sum_{i=1}^{4^k} f_i \ + \ w \ \sum_{j=1}^{\lambda} \theta_j} & (1 \le u \ \le \ 4^k) \\ \dfrac{w\theta_{u-\ 4^k}}{\sum_{i=1}^{4^k} f_i + w \ \sum_{j=1}^{\lambda} \theta_j} & (4^k \le u \ \ \le 4^k + \ \lambda) \end{cases}$$

where f_u (u = 1, 2, …, 4^k) is k-mer frequency being normalized to $\sum_{i=1}^{4^k} f_i = 1$ and λ is an integer, the number of the total counted tiers and w is a weight ranging in the interval of [0, 1] and θ_j computed as below equation:

$$\theta_j = \frac{1}{n-j-1} \sum_{i=1}^{n-j-1} \Theta\left(R_i R_{i+1}, R_{i+j} R_{i+j+1}\right) \ (j = 1, 2, \dots, \lambda; \ \lambda < n)$$

Where:

$$\Theta\left(R_i R_{i+1}, R_{i+j} R_{i+j+1}\right) = \frac{1}{\mu} \sum_{v=1}^{\mu} [P_v(R_i R_{i+1}) - \ P_v(R_{i+j} R_{i+j+1})]^2$$

where μ is the number of physicochemical indices in [13]. $P_v(R_iR_{i+1})$ and $P_v(R_{i+j}R_{i+j+1})$ are the numerical value of v^{th} physicochemical index for the dinucleotides R_iR_{i+1} and $R_{i+j}R_{i+j+1}$ at position i and position (i+j), respectively.

SC-PseTNC-General (stand for "General series correlation pseudo trinucleotide composition") [84]

Given a biological sequence S = $R_1R_2R_3 \dots R_n$ length of n, $R_i \in$ {A, C, G, T}. SC-PseTNC-General feature vector of sequence S is defined:

$$D = \{d_1, d_2, \dots, d_{64}, d_{64+1}, \dots, d_{64+\lambda}, d_{64+\lambda+1}, \dots, d_{64+\lambda\Lambda}\}$$

where d_u is calculated as following:

$$d_u = \begin{cases} \dfrac{f_u}{\sum_{i=1}^{64} f_i \; + \; w \sum_{j=1}^{\lambda\Lambda} \theta_j} & (1 \le u \; \le \; 64) \\ \dfrac{w\theta_{u-\,64}}{\sum_{i=1}^{64} f_i + w \sum_{j=1}^{\lambda\Lambda} \theta_j} & (64 + 1 \le u \; \le 64 + \; \lambda\Lambda) \end{cases}$$

where f_u (u = 1, 2, …, 64) is 3-mer frequency being normalized to $\sum_{i=1}^{64} f_i = 1$; λ is an integer, the number of the total counted tiers; w is a value in the range of [0, 1]; Λ is a the number of physicochemical indices; and θ_j computed as below equation:

$$\begin{cases} \theta_1 = \dfrac{1}{n-4} \sum_{i=1}^{n-4} J_{i,i+1}^{1} \\ \theta_2 = \dfrac{1}{n-4} \sum_{i=1}^{n-4} J_{i,i+1}^{2} \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \theta_\Lambda = \dfrac{1}{n-4} \sum_{i=1}^{n-4} J_{i,i+1}^{\Lambda} \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \theta_{\lambda\Lambda-1} = \dfrac{1}{n-\lambda-3} \sum_{i=1}^{n-\lambda-3} J_{i,i+1}^{\Lambda-1} \\ \theta_{\lambda\Lambda} = \dfrac{1}{n-\lambda-3} \sum_{i=1}^{n-\lambda-3} J_{i,i+\lambda}^{\Lambda} \end{cases}$$

The correlation function is defined:

$$\begin{cases} J_{i,i+m}^{v} = P_v(R_i R_{i+1} R_{i+2}) \,.\, P_v(R_{i+m} R_{i+m+1} R_{i+m+2}) \\ v = 1, 2, \ldots, \Lambda; i = 1, 2, \ldots, n-m-2 \end{cases}$$

where $P_v(R_iR_{i+1}R_{i+2})$ represents the numerical value of v^{th} (v = 1, 2, …, μ) physiochemical index for the trinucleotide $R_iR_{i+1}R_{i+2}$ at position i; and $P_v(R_{i+m}R_{i+m+1}R_{i+m+2})$ represents the numerical value of v^{th} (v = 1, 2, …, μ) physiochemical index for the trinucleotide $R_{i+m}R_{i+m+1}R_{i+m+2}$ at position i + m.

Bibliography

[1] "NCBI," National Center for Biotechnology Information, [Online]. Available: https://www.ncbi.nlm.nih.gov/genbank/statistics/. [Accessed 9 6 2016].

[2] Consortium UniProt, "UniProt: A hub for protein information," *Nucleic Acids Research,* vol. 43, no. D1, p. D204–D212, 2015.

[3] Z. Xing, J. Pei, and E. Keogh, "A brief survey on sequence classification," *ACM SIGKDD Explorations Newsletter,* vol. 12, no. 1, p. 40, 2010.

[4] I. Borozan, S. Watt, and V. Ferretti, "Integrating alignment-based and alignment-free sequence similarity measures for biological sequence classification," *Bioinformatics,* vol. 31, no. 9, p. 1396–1404, 2015.

[5] L. Chen and G. Guo, "Nearest neighbor classification of categorical data by attributes weighting," *Expert Systems with Applications,* vol. 42, no. 6, p. 3142–3149, 2015.

[6] M. J. Iqbal, I. Faye, B. B. Samir, and A. Md Said, "Efficient feature selection and classification of protein sequence data in bioinformatics," *The Scientific World Journal,* vol. 2014, p. 1–12, 2014.

[7] E. Weitschek, F. Cunial, and G. Felici, "LAF: Logic Alignment Free and its application to bacterial genomes classification," *BioData Mining,* vol. 8, no. 1, p. 39, 2015.

[8] T. H. Pham, T. B. Tran, T. B. Ho, K. Satou, and G. Valiente, "Qualitatively Predicting Acetylation and Methylation Areas in DNA sequences," *Genome Informatics,* vol. 16, no. 2, pp. 3-11, 2005.

[9] D. K. Pokholok, C. T. Harbison, S. Levine, M. Cole, N. M. Hannett, I. L. Tong, G. W. Bell, K. Walker, P. A. Rolfe, E. Herbolsheimer, J. Zeitlinger, F. Lewitter, D. K. Gifford, and R. A. Young, "Genome-wide map of nucleosome acetylation and methylation in yeast," *Cell,* vol. 122, no. 4, p. 517–527, 2005.

[10] M. Higashihara, J. D. Rebolledo-mendez, Y. Yamada, and K. Satou, "Application of a Feature Selection Method to Nucleosome Data : Accuracy Improvement and Comparison with Other Methods," *WSEAS TRANSACTIONS on BIOLOGY and BIOMEDICINE,* vol. 5, no. 5, p. 96–105, 2008.

[11] Nature Education, "Scitable," 2014. [Online]. Available: https://www.nature.com/scitable/topicpage/gene-expression-14121669. [Accessed 10 6 2017].

[12] a K. M. a Baten, B. C. H. Chang, S. K. Halgamuge, and J. Li, "Splice site identification using probabilistic parameters and SVM classification," *BMC Bioinformatics*, vol. 7, pp. S5-S15, 2006.

[13] S. H. Guo, E. Z. Deng, L. Q. Xu, H. Ding, H. Lin, W. Chen, and K. C. Chou, "INuc-PseKNC: A sequence-based predictor for predicting nucleosome positioning in genomes with pseudo k-tuple nucleotide composition," *Bioinformatics*, vol. 30, no. 11, p. 1522–1529, 2014.

[14] N. M. Berbenetz, C. Nislow, and G. W. Brown, "Diversity of eukaryotic DNA replication origins revealed by genome-wide analysis of chromatin structure," *PLoS Genetics,* vol. 6, 2010.

[15] W. Chen, L. Luo, and L. Zhang, "The organization of nucleosomes around splice sites," *Nucleic Acids Research,* vol. 38, p. 2788–2798, 2010.

[16] W. Chen, P.M. Feng, H. Lin, and K.C. Chou, "iSS-PseDNC: identifying splicing sites using pseudo dinucleotide composition," *BioMed Research International,* 2014.

[17] S. Schwartz, E. Meshorer, and G. Ast, "Chromatin organization marks exon-intron structure," *Nature Structural & Molecular Biology,* vol. 16, p. 990–995, 2009.

[18] T. Yasuda, K. Sugasawa, Y. Shimizu, S. Iwai, T. Shiomi, and F. Hanaoka, "Nucleosomal structure of undamaged DNA regions suppresses the non-

specific DNA binding of the XPC complex," *DNA Repair*, vol. 4, p. 389–395, 2005.

[19] N. Acır and C. Guzelis, "Automatic recognition of sleep spindles in EEG by using artificial neural networks," *Expert Systems with Applications*, vol. 27, no. 2004, p. 451–458, 2014.

[20] D. R. B. Stockwell, "LBS: Bayesian learning system for rapid expert system development," *Expert Systems with Applications*, vol. 6, no. 2, pp. 137-147, 1993.

[21] J. H. Min and Y. C. Lee, "Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters," *Expert Systems with Applications*, vol. 28, no. 4, p. 603–614, 2005.

[22] K. S. Shin, T. S. Lee, and H. J. Kim, "An application of support vector machines in bankruptcy prediction model," *Expert Systems with Applications*, vol. 28, no. 1, p. 127–135, 2005.

[23] Y. Zhang, C. Chu, Y. Chen, H. Zha, and X. Ji, "Splice site prediction using support vector machines with a Bayes kernel," *Expert Systems with Applications*, vol. 30, p. 73–81, 2006.

[24] V. Brendel and J. Kleffe, "Prediction of locally optimal splice sites in plant pre-mRNA with applications to gene identification in Arabidopsis thaliana genomic DNA," *Nucleic Acids Research*, vol. 26, no. 20, p. 4748–4757, 1998.

[25] L. Zhang and L. Luo, "Splice site prediction with quadratic discriminant analysis using diversity measure," *Nucleic Acids Research*, vol. 31, no. 21, p. 6214–6220, 2003.

[26] S. Degroeve, B. De Baets, Y. V. De Peer, and P. Rouze, "Feature subset selection for splice site prediction," *Bioinformatics*, vol. 18, p. S75–S83, 2002.

[27] C. Burge and S. Karlin, "Prediction of complete gene structure in human genomic DNA," *Journal of Computational Biology,* vol. 268, no. 1, p. 78–94, 1997.

[28] M. Pertea, X. Lin, and S. L. Salzberg, "GeneSplicer: A new computational method for splice site prediction," *Nucleic Acids Research,* vol. 29, no. 5, p. 1185–1190, 2001.

[29] D. J. Patterson, K. Yasuhara, and W. L. Ruzzo, "Pre-mRNA secondary structure prediction aids splice site prediction," in *Pacific Symposium on Biocomputing*, 2002.

[30] R. Weber, "DNA splice site prediction with kernels and voting," in *International conference on mathematical and engineering techniques in medicine and biological sciences*, Nevada, 2001.

[31] M. G. Reese, F. H. Eeckman, D. Kulp, and D. Haussler, "Improved splice site detection in genie," *Journal of Computational Biology,* vol. 4, no. 3, pp. 311-323, 1997.

[32] S. Sonnenburg, G. Ratsch, A. Jagota, and K. R. Muller, "New methods for splice site recognition," in *The international conference on artificial neural networks*, 2002.

[33] S. Sonnenburg, G. Schweikert, P. Philips, J. Behr, and G. Rätsch, "Accurate splice site prediction using support vector machines," *BMC Bioinformatics,* vol. 8, no. Suppl 10, p. S7, 2007.

[34] S. Degroeve, Y. Saeys, B. D. Baets, P. Rouzé, and Y. V. de Peer, "SpliceMachine: predicting splice sites from high-dimensional local context representations," *Bioinformatics,* vol. 21, no. 8, 2005.

[35] a K. M. a Baten, S. K. Halgamuge, and B. C. H. Chang, "Fast splice site detection using information content and feature reduction," *BMC bioinformatics,* vol. 9, no. Suppl 12, p. S8, 2008.

[36] P. K. Meher, T. K. Sahu, A. R. Rao, and S. D. Wahi, "A statistical approach for 5’ splice site prediction using short sequence motifs and without encoding sequence data," *BMC bioinformatics*, vol. 15, no. 1, p. 362, 2014.

[37] G. Yeo, and C. B. Burge, "Maximum entropy modeling of short sequence motifs with applications to RNA splicing signals," *Journal of Computational Biology*, vol. 1, no. 2-3, pp. 377-394, 2004.

[38] L. S. Ho and J. C. Rajapakse, "Splice site detection with a higher-order Markov model implemented on a neural network," *Genome Inform*, vol. 14, p. 64–72, 2003.

[39] P. K. Meher, T. K. Sahu, A. R. Rao, and S. D. Wahi, "A computational approach for prediction of donor splice sites with improved accuracy," *Journal of Theoretical Biology*, vol. 404, p. 285–294, 2016.

[40] M. Wang and A. Marín, "Characterization and prediction of alternative splice sites," *Gene*, vol. 366, p. 219–227, 2006.

[41] P. K. Meher, T. K. Sahu, A. R. Rao, and S. D. Wahi, "Identification of donor splice sites using support vector machine: a computational approach based on positional, compositional and dependency features," *Algorithms for Molecular Biology*, vol. 11, no. 1, p. 16, 2016.

[42] G. G. Towell, J. W. Shavlik, and M. O. Noordeweir, "Refinement of Approximate Domain Theories by Knowledge-Based Neural Networks," in *Proceeding of the Eighth National Conference on Artificial Intelligence*, 1990.

[43] G. Czibula, M. I. Bocicor, and I. G. Czibula, "Promoter sequences prediction Using Relational Association Rule Mining," *Evolutionary Bioinformatics*, vol. 8, pp. 81-196, 2012.

[44] H. Lin, E. Z. Deng, H. Ding, W. Chen, and K. C. Chou, "IPro54-PseKNC: A sequence-based predictor for identifying sigma-54 promoters in prokaryote

with pseudo k-tuple nucleotide composition," *Nucleic Acids Research,* vol. 42, no. 21, p. 12961–12972, 2014.

[45] A. Maleki, V. Vaezinia, and A. Fekri, "Promoter Prediction in Bacterial DNA Sequences Using Expectation Maximization and Support Vector Machine Learning Approach," *Data Mining in Genomics & Proteomics,* vol. 6, no. 2, 2015.

[46] B. Demeler and G. Zhou, "Neural network optimization for E. coli promoter prediction," *Nucleic Acids Research,* vol. 19, no. 7, 1991.

[47] L. G. Tavares, H. S. Lopes, and C. R. E. Lima, "A comparative study of machine learning methods for detecting promoters in bacterial DNA sequences," *Advanced Intelligent Computing Theories and Applications with Aspects of Artificial Intelligence,* pp. 959-966, 2008.

[48] S. de Avila e Silva, S. Echeverrigaray, and G. J. L. Gerhardt, "BacPP: Bacterial promoter prediction-A tool for accurate sigma-factor specific assignment in enterobacteria," *Journal of Theoretical Biology,* vol. 287, no. 1, p. 92–99, 2011.

[49] I. A. Shahmuradov, R. M. Razali, S. Bougouffa, A. Radovanovic, and V. B. Bajic, "Sequence analysis bTSSfinder : a novel tool for the prediction of promoters in cyanobacteria and Escherichia coli," *Bioinformatics,* vol. 33, no. 3, p. 334–340, 2017.

[50] K. Kouser, P. G. Lavanya, L. Rangarajan, and K. K Acharya, "Effective Feature Selection for Classification of Promoter Sequences," *PLOS one,* p. 1–20, 2016.

[51] X. Yi, Y.-D. Cai, Z. He, W. Cui, and X. Kong, "Prediction of nucleosome positioning based on transcription factor binding sites," *PLOS one,* vol. 5, no. 9, p. 1–7, 2010.

[52] K. Cartharius, K. Frech, K. Grote, B. Klocke, and M. Haltmeier, "MatInspector and beyond: promoter analysis based on transcription factor binding sites," *Bioinformatics,* vol. 21, p. 2933–2942, 2005.

[53] M. Tahir and M. Hayat, "iNuc-STNC: a sequence-based predictor for identification of nucleosome positioning in genomes by extending the concept of SAAC and Chou's PseAAC," *Molecular BioSystems*, vol. 12, p. 2587–2593, 2016.

[54] A. Awazu, "Prediction of nucleosome positioning by the incorporation of frequencies and distributions of three different nucleotide segment lengths into a general pseudo k-tuple nucleotide composition," *Bioinformatics*, vol. 33, no. 1, p. 42–48, 2017.

[55] M. Zakaria, M. AL-Shebany, and S. Sarhan, "Artificial Neural Network : A Brief Overview," *International Journal of Engineering Research and Applications*, vol. 4, no. 2, pp. 7-12, 2014.

[56] A. Castrounis, "KDnuggets," 2017. [Online]. Available: http://www.kdnuggets.com/2016/10/artificial-intelligence-deep-learning-neural-networks-explained.html. [Accessed 09 06 2017].

[57] S. Agatonovic-Kustrin and R. Beresford, "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research," *Journal of Pharmaceutical and Biomedical Analysis*, vol. 22, p. 717–727, 2000.

[58] "The data science blog," 11 08 2016. [Online]. Available: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/. [Accessed 09 06 2017].

[59] RSIP Vision Blogs, "RSIP Vision," [Online]. Available: http://www.rsipvision.com/exploring-deep-learning/. [Accessed 08 06 2017].

[60] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, San Francisco: Morgan Kaufmann, 2011.

[61] A. Bellili, M. Gilloux, and P. Gallinari, "An MLP-SVM combination architecture for offline handwritten digit recognition," *Digital Object Identifier*, vol. 5, p. 244–252, 2003.

[62] J. Inglada, "Automatic recognition of man-made objects in high resolution optical remote sensing images by SVM classification of geometric image features," *ISPRS Journal of Photogrammetry & Remote Sensing,* vol. 62, p. 236–248, 2007.

[63] M. Ikonomakis, S. Kotsiantis, and V. Tampakas, "Text Classification Using Machine Learning Techniques," *WSEAS TRANSACTIONS on COMPUTERS,* vol. 4, no. 8, pp. 966-974, 2005.

[64] C. Z. Cai, W. L. Wang, L. Z. Sun, and Y. Z. Chen, "Protein function classification via support vector machine approach," *Mathematical Biosciences,* vol. 185, p. 111–122, 2003.

[65] M. Halls-Moore, "QuantStart," 12 09 2014. [Online]. Available: https://www.quantstart.com/articles/Support-Vector-Machines-A-Guide-for-Beginners. [Accessed 08 06 2017].

[66] D. Benyamin, "CitizenNet Blog," 11 2012. [Online]. Available: http://blog.citizennet.com/blog/2012/11/10/random-forests-ensembles-and-performance-metrics. [Accessed 08 06 2017].

[67] C. Nguyen, Y. Wang, and H. N. Nguyen, "Random forest classifier combined with feature selection for breast cancer diagnosis and prognostic," *J. Biomedical Science and Engineering,* vol. 6, pp. 551-560, 2013.

[68] L. Breiman and A. Cutler, [Online]. Available: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm. [Accessed 08 06 2017].

[69] V. A. Kumar and N. Elavarasan, "A Survey on Dimensionality Reduction Technique," *International Journal of Emerging Trends & Technology in Computer Science,* vol. 3, no. 6, 2014.

[70] Y. Saeys, I. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics,* vol. 23, no. 19, p. 2507–2517, 2007.

[71] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers and Electrical Engineering,* vol. 40, pp. 16-28, 2014.

[72] "Juan Tapia Farias," [Online]. Available: https://jtapiafarias.wordpress.com/about/. [Accessed 08 06 2017].

[73] S. Kaushik, "Analytics Vidhya," 01 12 2016. [Online]. Available: https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/. [Accessed 08 06 2017].

[74] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining & Knowledge Management Process,* vol. 5, no. 2, 2015.

[75] I. H. Witten, E. Frank, and M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, San Francisco: Morgan Kaufmann, 2011.

[76] N. G. Nguyen, V. A. Tran, D. L. Ngo, D. Phan, F. R. Lumbanraja, M. R. Faisal, B. Abapihi, M. Kubo, and K. Satou, "DNA Sequence Classification by Convolutional Neural Network," *J. Biomedical Science and Engineering,* vol. 9, no. 9, p. 280–286, 2016.

[77] W. Chen, P. Feng, H. Ding, H. Lin, and K. C. Chou, "Using deformation energy to analyze nucleosome positioning in genomes," *Genomics,* vol. 107, no. 2–3, p. 69–75, 2016.

[78] X. F. Yi, Z. S. He, K. C. Chou, and X. Y. Kong, "Nucleosome positioning based on the sequence word composition," *Protein and Peptide Letters,* vol. 19, pp. 79-90, 2012.

[79] H. Schrem, J. Klempnauer, and J. Borlak, "Liver-Enriched Transcription Factors in Liver Function and Development. Part I: The Hepatocyte Nuclear Factor Network and Liver-Specific Gene Expression," *PHARMACOLOGICAL REVIEWS,* vol. 54, no. 1, p. 129–158, 2002.

[80] A. Liaw and M. Wiener, "Classification and Regression by randomForest," *R News*, vol. 2, no. 3, pp. 18-22, 2002.

[81] A. Karatzoglou, A. Smola, K. Hornik, and A. Zeileis, "kernlab - An S4 Package for Kernel Methods in R," *Journal of Statistical Software*, vol. 11, no. 9, pp. 1-20, 2004.

[82] L. G. Tavares, H. S. Lopes, and C. R. E. Lima, "A comparative study of machine learning methods for detecting promoters in bacterial DNA sequences," *Advanced Intelligent Computing Theories and Applications with Aspects of Artificial Intelligence*, pp. 959-966, 2008.

[83] H. D. Ismail, A. Jones, J. H. Kim, R. H. Newman, and D. B. KC, "RF-Phos: A Novel General Phosphorylation Site Prediction Tool Based on Random Forest," *BioMed Research International*, vol. 2016, 2016.

[84] W. Chen, T. Y. Lei, D. C. Jin, H. Lin, and K. C. Chou, "PseKNC: a flexible web server for generating pseudo K-tuple nucleotide composition," *Analytical biochemistry*, vol. 456, pp. 53-60, 2014.