Identifying Sigma 70 Promoters Using Multiple Windowing and Optimal Features

Md. Siddiqur Rahman
ID: 012 171 011

Department of Computer Science and Engineering

United International University

A thesis submitted for the degree of

Master of Science in Computer Science & Engineering

August 2018

Certificate

This thesis titled "Identifying Sigma70 Promoters Using Multiple Windowing and Optimal Features" submitted by Md. Siddiqur Rahman, Student ID: 012 171 011, has been accepted as Satisfactory in fulfillment of the requirement for the degree of Master of Science in Computer Science and Engineering on 19th September 2018.

Board of Examiners

1._____ Supervisor

Dr. Swakkhar Shatabda
Associate Professor & Undergraduate Program Coordinator,
Department of Computer Science and Engineering,
United International University, Dhaka–1212, Bangladesh

2._____ Head Examiner

Dr. Md. Abul Kashem Mia
Professor & Dean SoSE,
Department of Computer Science and Engineering,
United International University, Dhaka–1212, Bangladesh

3._____ Examiner-I

Dr. Mohammad Nurul Huda
Professor & Coordinator - MSCSE,
Department of Computer Science and Engineering,
United International University, Dhaka–1212, Bangladesh

4._____ Examiner-II

Ms. Rubaiya Rahtin Khan
Assistant Professor,
Department of Computer Science and Engineering,
United International University, Dhaka–1212, Bangladesh

5._____ Ex-Officio

Dr. Salekul Islam
Professor & Head of the Dept.,
Department of Computer Science and Engineering,
United International University, Dhaka–1212, Bangladesh

Declaration

This is to certify that the work entitled Identifying Sigma70 Promoters Using Multiple Windowing and Optimal Features" is the outcome of the research carried out by me under the supervision of Dr. Swakkhar Shatabda, Associate Professor & Undergraduate Program Coordinator, Department of Computer Science and Engineering, United International University, Dhaka-1212, Bangladesh.

Md. Siddiqur Rahman
Student ID: 012 171 011,
MSCSE student, Department of Computer Science & Engineering
United International University, Dhaka–1212, Bangladesh

In my capacity as supervisor of the candidates thesis, I certify that the above statements are true to the best of my knowledge.

Dr. Swakkhar Shatabda
Associate Professor & Undergraduate Program Coordinator,
Department of Computer Science and Engineering,
United International University, Dhaka–1212, Bangladesh

Abstract

In bacterial DNA, there are specific sequences of nucleotides called promoters that can bind to the RNA polymerase. Sigma70 (σ^{70}) is one of the most important promoter sequences due to it's presence in most of the DNA regulatory functions. In this thesis work, we have identified the most effective and optimal sequence-based features for prediction of σ^{70} promoter sequences in a bacterial genome. We have used both short-range and long-range DNA sequences in our proposed method. A very small number of effective features have been selected from a large number of the extracted features using different sized multi-window within the DNA sequences. We call our prediction method iPro70-FMWin and have made it freely accessible online via a web application established at `http://ipro70.pythonanywhere.com/server` for the sake of convenience of the researchers. We have tested our method using a standard benchmark dataset. In the experiments, iPro70-FMWin has achieved an area under the curve of the receiver operating characteristic and accuracy of 0.959 and 90.57% respectively which significantly outperforms the state-of-the-art predictors.

Keywords: sigma70 promoter, prokaryote, sequence-based features, multi-windowing, feature selection

Published Papers

Work relating to the research presented in this thesis has been published/submitted by the author in the following peer-reviewed journals and conferences:

1. Md. Siddiqur Rahman, Usma Aktar, Md. Rafsan Jani, and Swakkhar Shatabda, "Identifying Sigma70 Promoters Using Multiple Windowing and Minimal Features."Molecular Genetics and Genomics, 2018. DOI: `https://doi.org/10.1007/s00438-018-1487-5`.
   [**Accepted**]
2. Md. Siddiqur Rahman, Usma Aktar, Md. Rafsan Jani, and Swakkhar Shatabda, "iPromoter-FSEn: Identification of bacterial σ^{70} promoter sequences using feature subspace based ensemble classifier," Genomics, 2018. DOI: `https://doi.org/10.1016/j.ygeno.2018.07.011`.
   [**Accepted**]

Acknowledgements

First of all, I would like to thank almighty God, who has blessed and guide me so that I was able to accomplish my work. Working as a Masters student in United International University thesis was a magnificent as well as challenging experience to me. In all these period of my work, many people were directly or indirectly supporting me. It was hardly impossible for me to thrive in my work without the precious support of these personalities. Here is a small tribute to all those people.

In this very special occasion I would like to express my deep gratitude and appreciation to Dr. Swakkhar Shatabda, Associate Professor & Undergraduate Program Coordinator, Department of Computer Science and Engineering - United International University. I could not ask for more helpful and thoughtful supervisor, but there was many times where I had reached the 'crossroads' and each time he was there to steer me towards the right path. He listened to my problem and always made me feel as if my work mattered. Also I feel so good to say thanks to Usma Aktar, MSc student, United International University, who has given me her valuable time, advice, criticism and correction to my project from beginning up to the end of the working. In this very special moment, I would like to express my deepest thanks to United International University to give me financially supports as I have got 100% tution fee waiver and made me possible to finish my study.

Finally, I would like to thank my family to whom I owe a great deal. To my mother, thank you for inspiring me all the time for keeping me apart from depression. Also thanks to my elder brother, Abdur Razzak Ratan, he is also a great inspiration to me. Hence, great appreciation and enormous thanks are due to him, for without his supporting, I am sure this thesis would never have been completed. I thank you all.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

DNA	**D**eoxyribo**N**ucleic **A**cid
RNA	**R**ibo**N**ucleic **A**cid
ROC	**R**eceiver **O**perating **C**haracteristic
ANN	**A**rtificial **N**eural **N**etwork
SVM	**S**upport **V**ector **M**achine
LR	**L**ogistic **R**egression
KNN	**K**-**N**earest **N**eighbor
DTC	**D**ecision **T**ree **C**lassifier
GNB	**G**aussian **N**aive **B**ayes
LDA	**L**inear **D**iscriminant **A**nalysis
E. coli	***E**scherichia **c**oli*
bp	**B**ase **P**air
TSS	**T**ranscript **S**tart **S**ite
A	**A**denine
C	**C**ytosine
G	**G**uanine
T	**T**hymine
DNase	**D**eoxyribo**N**uclease
RBF	**R**adial **B**asis **F**unction

Acc	**A**ccuracy
S_n	**Sen**sitivity
S_p	**Sp**ecificity
auPR	**A**rea **U**nder **P**recision-**R**ecall Curve
auROC	**A**rea **U**nder **R**eceiver **O**perating **C**haracteristic curve
MCC	**M**athews **C**orrelation **C**oefficient
TP	**T**rue **P**ositive
TN	**T**rue **N**egative
FP	**F**alse **P**ositive
FN	**F**alse **N**egative

Chapter 1

Introduction

The DNA transcription is very important event and it is a part of the Central Dogma of biology. DNA contains every information that is needed to be coded. But from the DNA, information should be transfered to some form which can actually work because DNA itself is not working. It is the master and rule our whole cell by sitting the inside of nucleus.

Now, what DNA actually does? It provides some messengers which go there and follow the task that the DNA have to do. And those messengers are the RNA. Therefore, RNA contains the information those are coded in the DNA. After that, RNA moves from nucleus into the cytosol. Then, the RNA is translated into protein products. Those proteins are the functional unit and they do all the functions according to the law of DNA whatever is written inside the DNA. In Figure 1.1 is representing the overview of DNA transcription. In DNA, promoters are binding sites for RNA polymerase. In other word, promoters are the sequence of several nucleotides, which are essential for the initialization and regulation of gene transcription. Several key factors are involved in the process of gene transcription. In bacteria, sigma (σ) factor is a subunit of RNA polymerase and one of the most contributor for recognizing and binding promoter sequence during gene transcription (Figure 1.2).

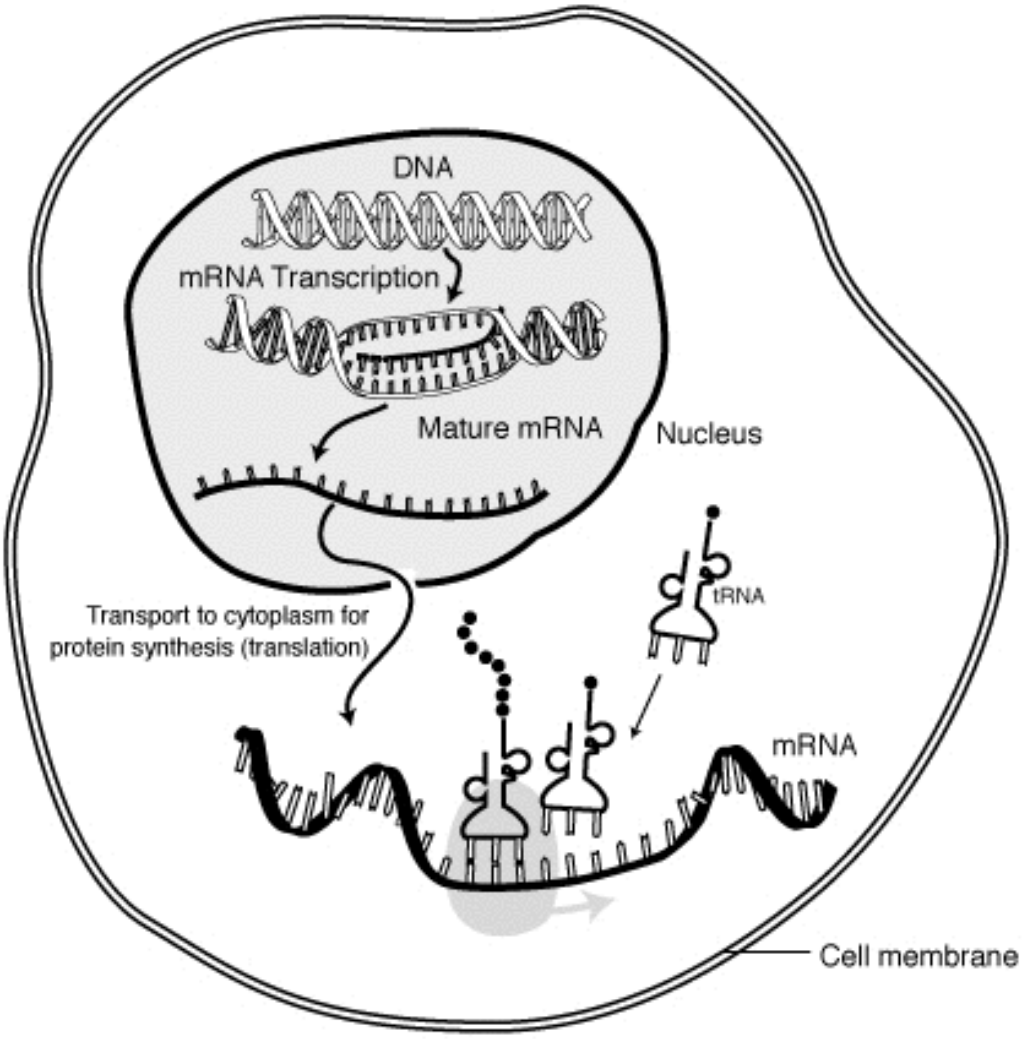


Figure 1.1: RNA is transcribed in the nucleus. Image Source: [1]

To start DNA transcription initialization, it is required the signal and the protein. For instance RNA polymerase. The important property of RNA polymerase is that it can initiate by adding nucleotide sequence on its own. There are several sigma (σ) factors for recognizing different promoters sequences, for instance, $\sigma^{19}, \sigma^{24}, \sigma^{28}, \sigma^{32}, \sigma^{38}, \sigma^{54}$ and σ^{70}. Among them σ^{70} factor also, called "housekeeping" sigma factor or primary sigma factor, that is responsible for the regulation of the transcription of most genes in growing cells [9].

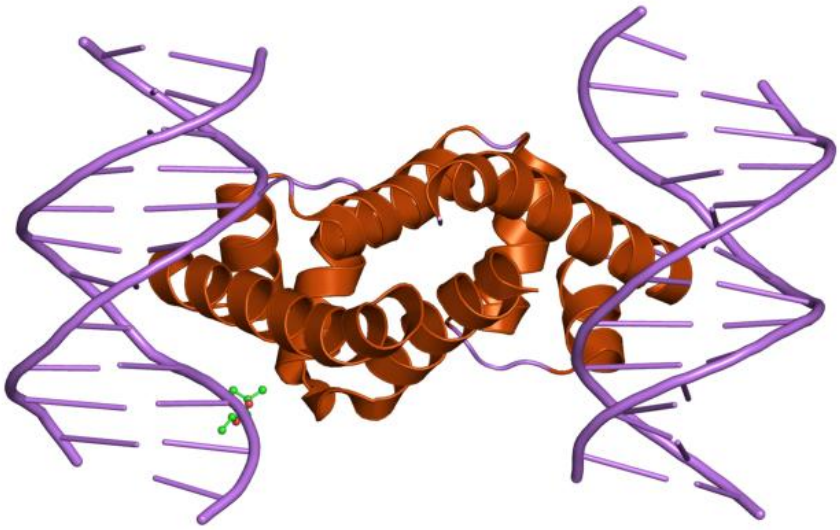


Figure 1.2: Crystal structure of Escherichia coli sigma70 region 4 bound to its -35 element DNA. Image Source: [2]

Figure 1.3 illustrates the process how sigma (σ) factor helps RNA polymerase to recognize promoter sequence. Initially, RNA polymerase is binding a random place into the DNA sequence but it is not detecting the promoter sequence very well. So the σ factor comes and attaches with RNA polymerase to find the promoter region. After that, the polymerase changes its structure kind of or modify its structure a little bit. Because upon binding to the promoter segment, it stretches itself when it is bound with σ factor. Here, the important thing is that σ factor is required to designate and determine the consensus sequence of promoter. But the RNA polymerase is bound with σ factor and they cannot start or initiate transcription or cannot move along the DNA. So during this part the only way is to detect where the promoter is. Now it needs to synthesize the transcript. So, in this case, it needs to dissociate the σ factor before it starts transcription. After releasing the σ factor, the structure of the RNA polymerase shift back to the original form. Then, the RNA polymerase starts DNA transcription.

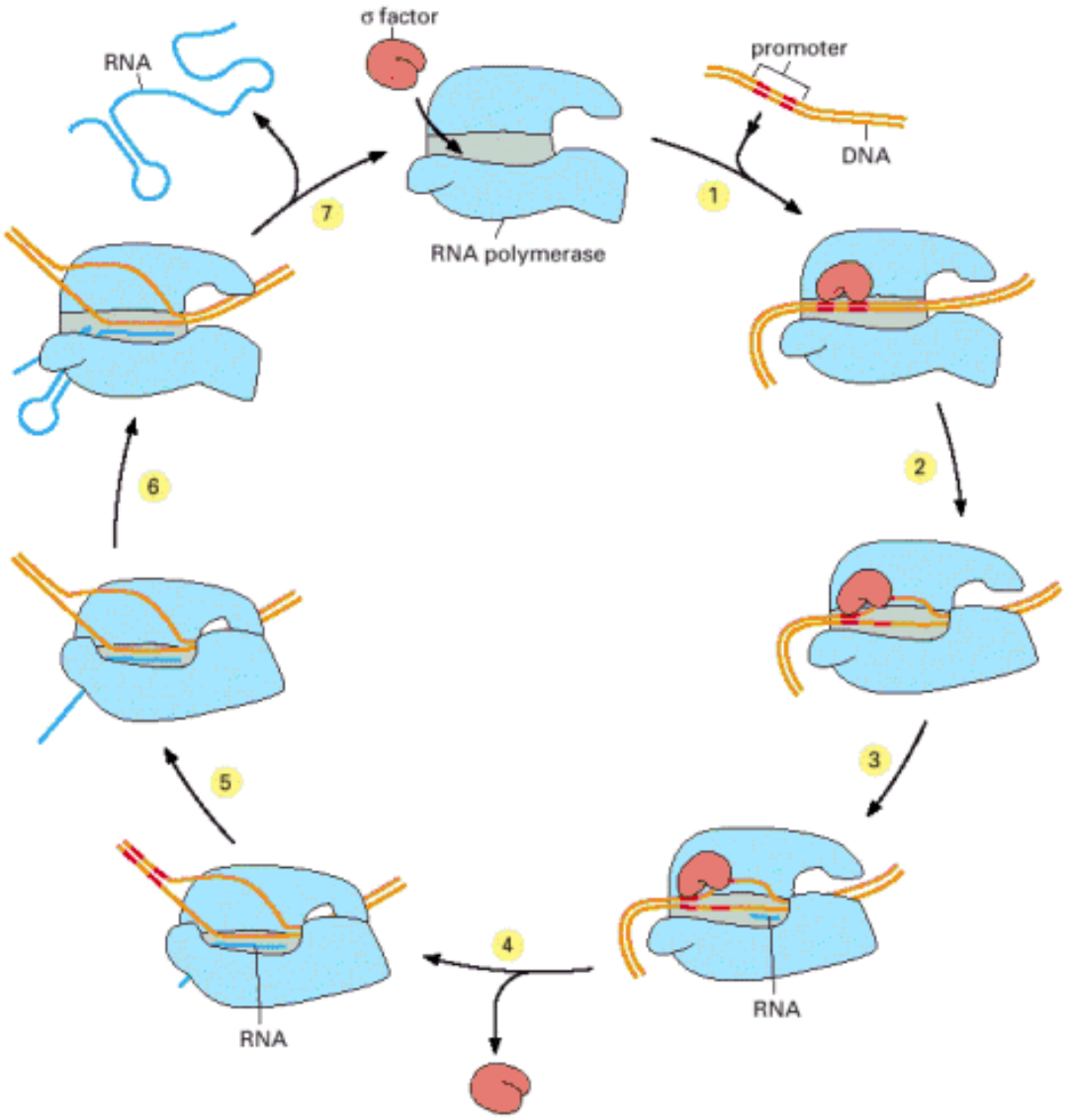


Figure 1.3: The Sigma Factor is involved in the initiation process of prokaryotic transcription. Image Source: [3]

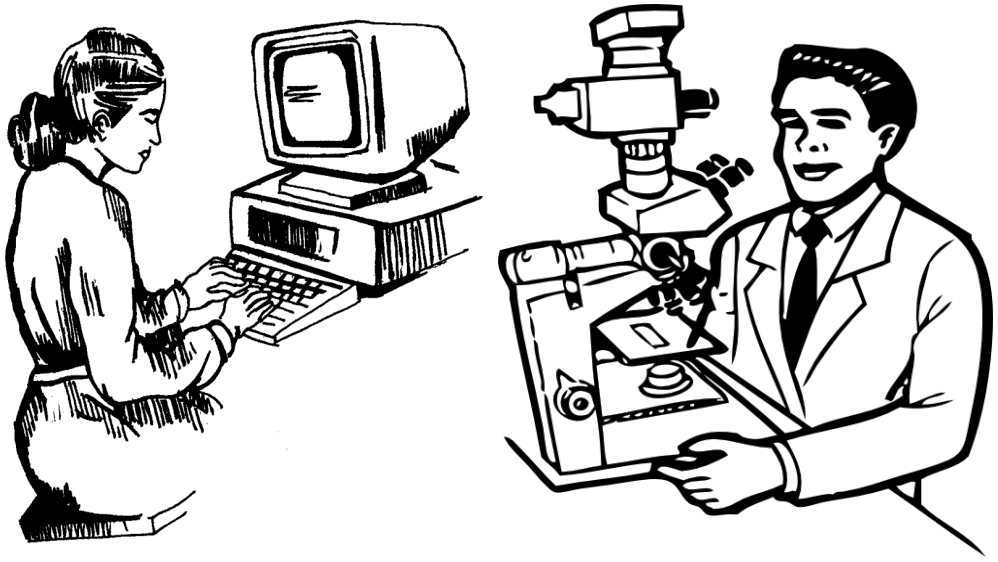
Specific sigma factor is responsible to regulate specific gene transcription. Therefore, promoter sequences are defined by the name of sigma factor. Figure 1.2 shows sigma70 promoter region of *Escheria Coli*. With the growth of available promoter sequences from previous laboratory experiments, computational algorithms are becoming popular instead of wet-experimental approach. This is due to the time-consuming and expensive nature of laboratory methods [10] compared to faster and cost-effective com-

putational methods. A good number of machine learning algorithms have been used to develop computational tools for prediction methods in the literature. They have included Support Vector Machines (SVM) [11, 12], Artificial Neural Networks (ANN) [13–15], Markov Models [16], Hidden Markov Models [17] and Random Forests [18]. Apart from machine learning methods position weight matrices [19] and phylogenetic foot-printing [20] have been used for prediction of promoter sequences.

In this study, we have used sequence-based features for prediction of sigma 70 promoter sequences. The main motivation of this work is to rely on the hypothesis that the information about the promoter sequences must be there within the DNA sequences. However, we have included a number of features which extracted from the sequences that include different statistical measures, position-specific counts, k-mer compositions, gapped compositions, etc. In addition, we have applied multiple window-based feature extraction to enhance the features. We have performed an analysis on the features to select an optimal set of features. Based on the optimal set of features those have been found in our experiments, we have proposed a novel predictor named, iPro70-FMWin. Besides, we have used a standard benchmark dataset to test the performance of iPro70-FMWin. Our proposed method significantly outperforms the state-of-the-art methods in terms of standard statistical measures. For the practical testing purpose, a web-server has been developed for identification of σ^{70} promoter sequences and can be freely accessible at `http://ipro70.pythonanywhere.com/server`.

1.1 Motivation

This section discusses the three factors that motivate the research undertaken in this study. First, identifying promoter sequence was very slow process and expensive in laboratory by doing wet-experimental approach while computational approach is much more faster than the wet-experimental approach as well as cost effective. Second, we have used multiple window approach for different short-range and long-range sequences to find out the most effective and influential regions within the DNA segment. Finally, we have tried to optimize the features based on their predictive accuracy to make our algorithm faster in terms of training and testing.



(a) **(b)**

Figure 1.4: Figures are showing the approaches for identifying promoter sequence. (a) Computational approach [4] and (b) Wet-experimental approach [5].

As we had available promoter sequences from previous experiments in laboratory, therefore in this study we went for the computational approach to solve a biological problem.

1.2 Objectives of the Thesis

In this work, the main objective is to identify sigma70 (σ^{70}) promoters focusing on multiple windowing within the DNA segments for feature extraction and also find out the optimal number of features by feature selection. Another objective is to compare among multiple machine learning algorithms. Finally, to develop an iterative real-time web-server named iPro70-FMWin for identifying σ^{70} promoter which can be freely accessible.

1.3 Thesis Contributions

In this work, we mainly focused on both short-range of DNA segment as well as long-range of DNA segments using multiple windowing approaches. According to our goal of

using multiple windowing, we have selected different region of the given DNA segments. After many experiments, we have tuned our window for the different places to extract features. The main logic, behind using multiple windowing, was to find out a optimal number of most influential features. We have also implemented a real-time web server for the future researcher to test their promoter sequences.

After getting the dataset, we have applied our feature extraction method where we have used our multiple windowing approaches. Then, we have tried six popular machine learning algorithm to find out the best classifier in terms of their performance. Then, our feature selection method have been applied to find out a optimal number of most effective features based on their predictive ability. Finally, we have implemented an interactive web server. All the implementation was done by using python and python Flask has been used for the web server.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows:

Chapter 2 follows up many related works in order to highlight some of the typical approaches and computational studies on promoters.

Chapter 3 is mainly focusing on materials and methods those have been used in this work. It also describes the formulation of DNA sequencess, feature extraction using different approaches and feature selection method. After that, it describes some popular machine learning algorithm and some measures to evaluate performance.

Chapter 4 mainly describes the experimental analysis where we have divided the whole experimental process into three different approaches. In this chapter, the first typical analysis approach is shown with graphical representation with results table such as flow-chart, box-plot, and ROC. Secondly, the experiment is shown using a multi-windowing approach. Finally, a feature selection algorithm is applied to the multi-windowing approach. Also, the same graphical representations and results table are shown for the second and final approaches. In this chapter, we also describe the best 27 features those are in the leading stage to identify

sigma70 promoter sequences compared with some popular state-of-the-art algorithms. In the end, we represent our designed web-server interface.

Chapter 5 concludes the thesis with future works.

Chapter 2

Related Work

In this chapter, we have described many computational approaches about DNA analysis including typical approaches. A descriptions have been given on various state-of-the-art techniques those were found in the literature for solving DNA promoter identification problem. Most of the techniques have been considered only for either short-rage or long-range of promoter sequences. Also, a small number of dataset has been used for training and testing. Many methods have been proposed for the identification of promoter sequences such as Neural Networks [13], Sequence alignment kernel trick [21] which was proposed to identify σ^{70}, scoring matrix [22], support vector machines [11], and Random Forest Classifier [18].

2.1 Typical Literature

The first influence of the DNA transformation was sparked by Griffith in 1928 [23]. In this work, he invented properties called trait, where it was a unique form of a prototypical characteristic of the organism. For instance, one of the characteristic of an organism is eye color while example for traits are blue, brown and hazel.

After that, there were a massive work was done on the basis of DNA transforming principle. For instance, DNA transforming principle was identified by the Avery-MacLeodMcCarty experiment in 1869 [24, 25]. Also, in 1869, Swiss physician Friedrich Miescher first extracted DNA from the cell, when he was working as a bacteriologist from the pus in surgical bandages on bacteria. The molecule he called *nuclein* because it was found in the nucleus [26]. In the late 1970s and early 1980s, with the intro-

duction of the DNA test, scientists observed the potential for more powerful testing for identification and determination of biological relationships. It was found in 1984, a long polymer made from a repeat unit called DNA nucleotide [27]. After more than a decade, scientists were able to recognize promoter sequence in DNA [28]. All of these typical processes were time consuming as well as expensive. The next section will describe about computational approach that rapidly changed the world in the field of Bioinformatics.

2.2 Computational Literature

The first application of computational approach was Neural Networks for identifying promoter sequences in [13]. A set of 80 known promoter sequence was used to train the neural network by combining those sequences with different numbers of random sequences. The conserved -10 region and -35 region of the promoter sequences and a combination of these regions were used in three independent training sets. The prediction accuracy of the resulting weight matrix was tested against a separate set of 30 known promoter sequences and 1500 random sequences. The effects of the network's topology, the extent of training, the number of random sequences in the training set and the effects of different data representations were examined and optimized. In this work, the author considered about only 80 known promoter sequences where it is a high probability that the algorithm could make over-fitting.

A Sigma70 (σ^{70}) promoter sequence predictor was proposed in [21] which is kernel based sequence alignment. They trained their method on 669 experimentally validated *Escherichia coli* promoter sequences compared to only 80 promoter sequences trained by the former method. In this study, the author proposed a new method for recognition of prokaryotic promoter regions with start-points of transcription. The method is based on Sequence Alignment Kernel, a function reflecting the quantitative measure of match between two sequences.

An analysis on σ^{70} sequences was performed in [22]. Authors in [22] used position correlation scoring matrix in their work. The work was based on the conservation analysis of the 683 latest experimentally verified σ^{70} promoter sequences of Escherichia coli K-12. It was found that the conservative hexamers segments in different sites play a

key role of promoter regions. Therefore, they claimed in their approach that it is a novel position-correlation scoring matrix (PCSM) algorithm for predicting σ^{70} promoter.

Pro54BD, an experimentally verified database was proposed in [29] that contained 210 experimentally verified σ^{54} sequences. Lin et al. [11] used support vector machines to predict σ^{54} promoter sequences and proposed iPro54-PseKNC. iPro70-PseZNC was proposed in [12] for σ^{70} promoter sequence detection using support vector machine and pseudo nucleotide composition. Liu et al. [18] proposed iPromoter-2L which is a two-layer promoter sequence detector. In the first layer, a classifier is used to identify the promoter sequences from non-promoter regions and then in the second layer it distinguishes between several types of sigma promoter sequences. They have used Random Forest Classifier to predict different sigma factors.

2.3 Summary

This chapter has discussed different methods or techniques those are found in the literature for DNA analysis and also has described for identifying promoter sequence. In the next chapter is organized about the Materials and Methods that have been used to generate the feature set for training and testing our methodologies.

Chapter 3

Materials and Methods

The methodology of the paper has been given in this section which was actually suggested by K.C. Chou as five steps for developing methods for prediction of attributes of biological entities [30, 31]. Those five steps are: i) the description of the benchmark dataset use; ii) representation of the sample or instances; iii) description of the classification algorithms; iv) performance evaluation techniques and lastly v) development of a web server. The rest of the section is organized in the similar fashion.

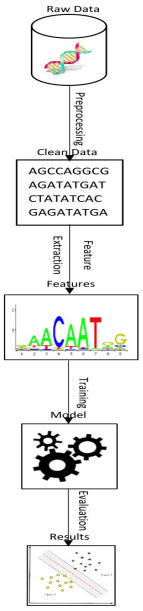


Figure 3.1: System Diagram

3.1 Benchmark Dataset

The first step in developing a prediction method is to construct a benchmark dataset for the classification problem. In this work, a high-quality pre-constructed benchmark dataset has been collected from a reliable data source: `http://lin-group.cn/server/iPro70PseZNC/data.html` [12]. The raw data has been collected from RegulonDB 9.0 (`http://regulondb.ccg.unam.mx/` [32]). There were total 2141 DNA sequences. We have used cross-validation on those total dataset to train and test our method. Among them 741 were σ^{70} promoters sequence of *E. coli* K-12. Rest of them were randomly chosen non-promoter sequences which were extracted from coding regions and intergenic regions of *E.coli* K-12 genome. All of them were 81 bp where 60 bp upstream and 20 bp downstream of the TSS (Transcript Start Site) at each sequence in the dataset. The constructed dataset was slightly imbalanced among promoters and non-promoters (ratio 1:1.89) and which can be expressed by

$$\mathbb{S} = \mathbb{S}^{+} \cup \mathbb{S}^{-} \tag{3.1}$$

where $\mathbb{S}^{+}$ represents positive samples or promoter DNA sequences, $\mathbb{S}^{-}$ represents negative samples or non-promoter DNA sequences, and the symbol $\cup$ represents the union in the set theory.

3.1.1 Formulation of DNA Samples

A DNA sequence is the combination of four nucleic acids named **A**denine (A), **G**uanine (G), **C**ytosine (C), and **T**hymine (T). Since a machine learning algorithm can recognize only numerical features as input, therefore we needed to formulate the DNA sequences with an effective mathematical illustration for feature extraction. The straightforward representation of a DNA segment with its entire nucleic acid residues can be formulated as

$$D = R_1 R_2 R_3 R_4 R_5 R_6 \ldots\ldots\ldots R_L \tag{3.2}$$

where each nucleic acid residue is represented as $R_{i(1,2,3,4,\ldots,L)}$ at the position of i in the DNA sequence where L the length of that DNA sequence. Typically, the input requirement of a machine learning algorithm is a numerical representation of objects for the statistical prediction analysis. Now, the first step is how can we express DNA

sequence into a numerical representation as an input for the machine learning algorithm to get as much as a possible accuracy of prediction.

Last two decades, many exciting methods have been invented which has already done a thrilling result in predicting promoter sequences [11, 12, 33, 34]. In this work, we have focused on multiple window approach to extract features. Detail about feature extraction is shown in bellow.

3.1.1.1 Statistical Measures

Among the statistical measures, first, we have included a simple frequency count of each nucleotide along through the 81 bp DNA segment. One of the other important features was the GC content [35] with a high ranked in promoter predicting score. We have also incorporated three common statistical measures. Those are standard deviation, mean, and variance. Low standard deviation indicates that the data distribution is closed to the mean of the distribution whereas the high standard deviation indicates spreading range of the data distribution. In addition, according to the previous research, there should be an equal frequency of the four DNA bases (A, C, G, T) [36] if there is no mutational or selective pressure. Therefore we also have used GC-Skew [37] for the whole sequence.

3.1.1.2 *k*-mer Composition

k-mer stands for the k-length substring of all possible combination of A, C, G, and T through the full string of the given DNA segment. k-mer is effectively used in the field of computational biology such as sequence assembly, sequence alignment, and variability in human genome mutation rate explanation [38–40]. In this work, we have constructed features using k-mer where k = 2,3,4,5,6. The frequency or composition of the k-mer was taken as features.

3.1.1.3 g-gapped k-mer composition

In addition with k-mer compositions, we have used g-gapped di-nucleotide compositions. However, we also have extend the idea of di-nucleotides to tri-nucleotides as well and hence towards a possible generalization of any k-mers. We have considered gaps, $g = 1, 2, 3, \cdots, 24$. More detail about this feature is shown in Table 3.1.

Table 3.1: k-mer composition using g-gap.

k	g	Pattern
2	1	X_X
2	2	X__X
2	3	X___X
...	...	.....
...	...	.....
2	24	X_____X

When constructing the features for k-mers where $k = 3$, we have used patterns like X_XX and XX_X.

3.1.1.4 Pattern Finding

It has been proven that there are some patterns of nucleotide sequences those are very important to recognize promoter region in DNA sequence [41–43]. Some of them have been used in this work for feature extraction such as TATAAT, TAATAT, TATAAA, AAATAT, TTGACA, ACAGTT, and AACGAT. We tried to exact or approximate pattern matching for all of them and also their right shifted cyclic form except the last one. The string matching of each pattern has been counted if there was at least three matching. Otherwise, the counted value was considered as a zero.

3.1.1.5 Positioning Distance Count

In this paper, we also have consider the total summation of each nucleotides positioning distance. The calculation of this feature is illustrated in Figure 3.2. For this example, total summation of positioning distance for A, C, G, and T are 9, 4, 4, and 4 respectively.

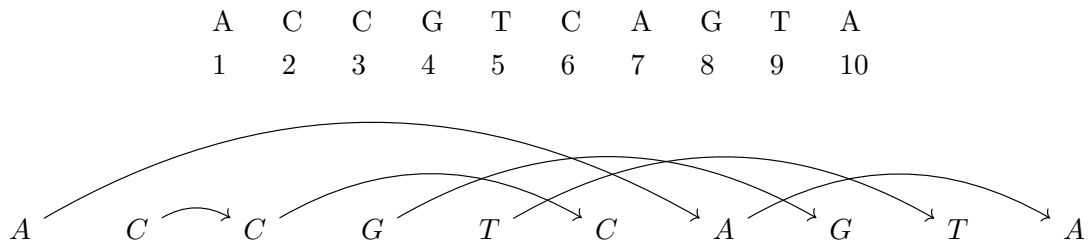


Figure 3.2: Illustration of positioning distance counts feature.

Identification of promoter regulatory regions can be effectively found with a high prediction scores by another popular approach called DNase I hypersensitive sites sequencing [44–46]. Similarly, we have used Dinucleotide Parameters Based on DNaseI Digestion Data in this article.

3.1.1.6 Window Approach

Typically, most of the previous researchers have used the full length of the given DNA segment for feature extraction. In this study, window method has been applied to the above described all the features group to get an extra benefit. Multi-window approach was used successfully to identify sigma70 promoter by Hao Lin in [12]. However, different window size was for the different feature group. Window size have been selected different in length because promoters are not fixed in length. Details of the window approach are shown in Table 3.2.

Table 3.2: Applying windowing on all the above described features

Group	Feature Name	Window Size	Total
1	Frequency count of each nucleotides and	-60 bp to +20 bp	5
	(G + C) content	-60 bp to -51 bp	5
		-50 bp to -41 bp	5
		-40 bp to -26 bp	5
		-15 bp to -6 bp	5
		-5 bp to -1 bp	5
		0 bp to -9 bp	5

		10 bp to 20 bp	5
2	Mean, Variance, Standard Deviation	-60 bp to 20 bp	3
		-15 bp to -6 bp	3
3	GC-Skew	-60 bp to 20 bp	81
4	k-mer, k=2,3,4,5,6	-60 bp to 20 bp	5456
	$N^2 + N^3 + N^4 + N^5 + N^6 = 5456$	-40 bp to -26 bp	5456
	N=4 for A, C, G, and T	-15 bp to -6 bp	5456
5	g-gap into k-mer, g=1,2,3,....,24 and k=2, $16 \times 24 = 384$	-60 bp to 20 bp	384
	g-gap into k-mer, g=1,2,3,....,11 and k=2, $16 \times 11 = 176$	-40 bp to -26 bp	176
	g-gap into k-mer, g=1,2,3,.....,7 and k=2, $16 \times 7 = 112$	-15 bp to -6 bp	112
6	g-gap into k-mer, g=1,2,3,....,24 and k=3, $64 \times 24 \times 2 = 3072$	-60 bp to 20 bp	3072
	g-gap into k-mer, g=1,2,3,....,11 and k=3, $64 \times 11 \times 2 = 1408$	-40 bp to -26 bp	1408
	g-gap into k-mer, g=1,2,3,.....,7 and k=3, $64 \times 7 \times 2 = 896$	-15 bp to -6 bp	896
7	Approximate Cyclic Right Shifted Pattern Count: TATAAT, TAATAT, TATAAA, AAATAT, $6 \times 4 = 24$	-15 bp to -6 bp	24
8	Approximate Cyclic Right Shifted Pattern Count: TTGACA and ACAGTT, $6 \times 2 = 12$	-40 bp to -26 bp	12
9	Approximate Pattern Count: AACGAT	1 bp to 6 bp	1
10	Positioning Distance Count for A, C, G, and T	-60 bp to 20 bp	4
		-40 bp to -26 bp	4
		-15 bp to -6 bp	4
11	Dinucleotide Parameters Based on DNaseI Digestion Data	-60 bp to 20 bp	1
		-40 bp to -26 bp	1
		-15 bp to -6 bp	1
			Total: 22595

3.1.2 Feature Selection

The feature selection strategy typically chooses a small number of useful features from a large number of unnecessary or irrelevant features. This is broadly used to achieve the most important and short sized subsets of total features [47]. Practical experiences have proven that more effective prediction tools are achievable by feature selection. There

are many feature selection methods in computational approach. Kira and Rendell developed an algorithm in 1992 called Relief that takes a filter method for feature selection [48]. There are many other machine learning algorithms those are widely used for feature selection such as AdaBoost Gabor feature selection [49], combining SVM for feature selection [50], a comparative study using Random Forest on feature selection [51], and Logistic Regression [52]. In our study, we have implemented the feature selection strategy for the following four aspects: (i) model simplification [53], (ii) reduce training time, (iii) avoid dimensionality problem, and (iv) reduce over-fitting [54]. In order to full-fill those four aspects, we have used the AdaBoost algorithm to find the best features based on their predictive accuracy [49].

3.1.2.1 AdaBoost Training For Feature Selection

In order to improve prediction performance, the AdaBoost algorithm is often used with other types of machine learning algorithms. Initially, a simple classifier has been fitted on the data also called a decision stump which splits the data into just two regions. Then, the class correctly classified will be given less weight edge in the next iteration and higher weight edge from misclassified classes. After that, another decision stump or weak classifier has fitted on the data and has changed the weights again for the next iteration. Here check the misclassified for which weight has been increased once it finishes the iterations these are combined with weights. Also, weights are automatically calculated for each classifier at each iteration based on error rate to come up with a strong classifier which predicts the classes with surprising accuracy. Let's have a look at the equation for AdaBoost,

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \lambda_j \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

where $h_j(x)$ is a simple threshold function consisting of only one simple feature $f_j(x)$, λ_j is a threshold and p_j is a parity to indicate the direction of the inequality. The threshold value is determined by the mean value of the positive samples and the mean value of the negative samples on the jth feature response:

$$\lambda_j = \frac{1}{2}\left(\frac{1}{m}\sum_{p=1}^{m} f_j(x_p|y_p = 1) + \frac{1}{l}\sum_{n=1}^{l} f_j(x_n|y_m = 0)\right) \tag{3.4}$$

More details about this procedure are to be found in [49]. In this study, we have performed the AdaBoost algorithm to identify the features which have most prediction capability. Finally, we have found a optimal number of effective features with a great accuracy. To find the best feature set, we have applied AdaBoost using 10-fold cross-validation on our feature set. Then, it has selected different features in each fold as the best features. We have just tried to count their frequency based on their presence in each fold. After, we have found only 27 features among 22595 features (Shown in Table **??**) which were present in all 10-folds. Then, we have trained our model by only those 27 features and have tested by the new test set.

3.2 Classification Algorithms

Choosing classifiers or model is crucial to get a desirable outcome. One of the main focusing points in this work was to choose an effective classification model for our supervised learning task. For picking up the perfect classifier, we have tried different classifiers on our sample data such as Support Vector Machine (SVM) [6], Logistic Regression (LR) [55], K-Nearest Neighbor (KNN) [7], Decision Tree Classifier (DTC) [56], Gaussian Naive Bayes (GNB) [8] and Linear Discriminant Analysis (LDA) [57]. After that, we have summarized them in one classifier. In this section, we are going to discuss a brief on them.

3.2.1 Support Vector Machine

Support Vector Machine (SVM) [6] is a supervised learning classifier that tries to maximize the margin between two classes by mapping input data instances to a higher dimensional space. In other words, it looks at the extremes of the dataset and draws a decision boundary also known as a hyperplane near the extreme points in the dataset. We have performed classification by finding the hyperplane that differentiates the two classes. Essentially, kernel function has been used to transform nonlinear into a linear space. There are some popular kernel functions to transform data into high dimensional feature space such as polynomial kernel, radial basis function, sigmoid kernel etc. Choosing the correct kernel function is a non-trivial task. A popular parameter choosing technique k-fold cross-validation have been used to choose our parameter. We

have used the radial basis function (RBF) kernel to overcome the problem with transformation into higher dimensional feature space. The downside of SVM is the training time much longer as it's much more computationally intensive.

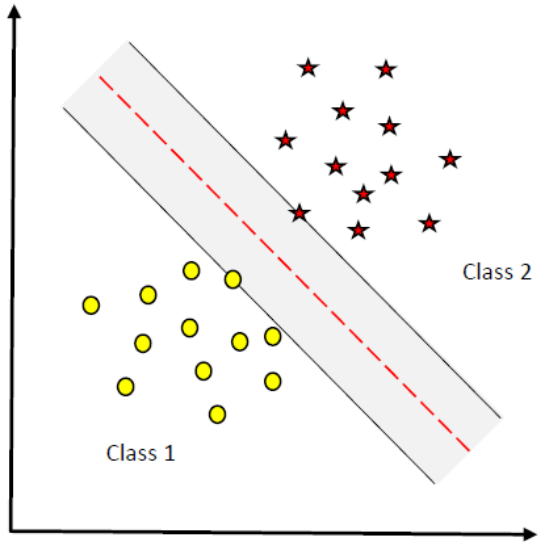


Figure 3.3: Support Vector Machine [6]

3.2.2 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) [57] is most commonly used as a dimensionality reduction technique in the preprocessing step for data classification in machine learning applications. The goal is to project a dataset into a lower dimensional space with good class separability in order to avoid over-fitting. Logistic Regression (LR) [55] is another popular technique borrowed by machine learning from the field statistics.

3.2.3 Logistic Regression

Logistic Regression is similar to linear regression in a sense that they both have the same goal of estimating the values of parameters coefficients. As a result, at the end of the training of the machine learning model, we have got the function that best describes the relationship between the non-input and the output values, unlike linear regression the prediction of the output is transformed using a nonlinear function called the logistic function. Some variations of the same are called the sigmoid function as well as the logit function.

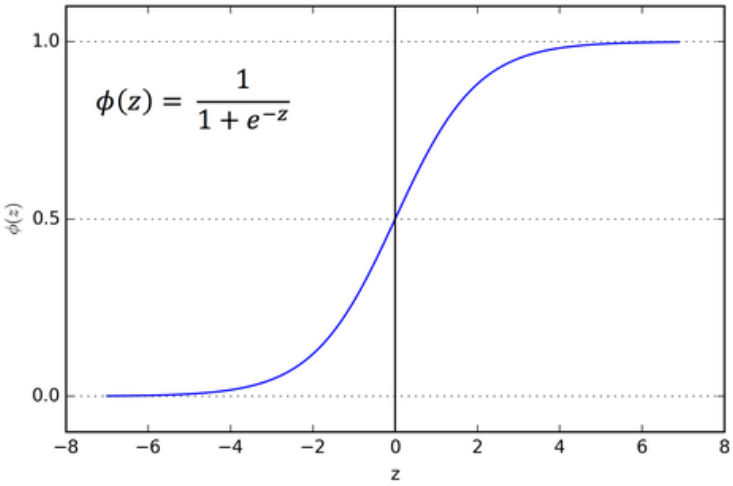


Figure 3.4: Logistic Regression

Looking at the logistic regression equations. The logistic regression hypothesis is defined as,

$$h_\theta(x) = g(\theta^T x) \tag{3.5}$$

where $h_\theta(x)$ is a function of $g(\theta^T x)$ and function $g()$ is the sigmoid function just define as,

$$g(z) = \frac{1}{1+e^{-z}} \tag{3.6}$$

3.2.4 K-Nearest Neighbor

In pattern recognition, K-Nearest Neighbor (KNN) [7] algorithm is a method for classifying the object based on the closest training example in the feature space. KNN is a type of instance-based learning where the function is only approximated locally and all competition is delayed until classification. The KNN algorithm is fundamental and one of the simplest classification technique when they little or no prior knowledge about the distribution of the data. The K in KNN refers to the number of nearest neighbors that the classifier will use to make its prediction.

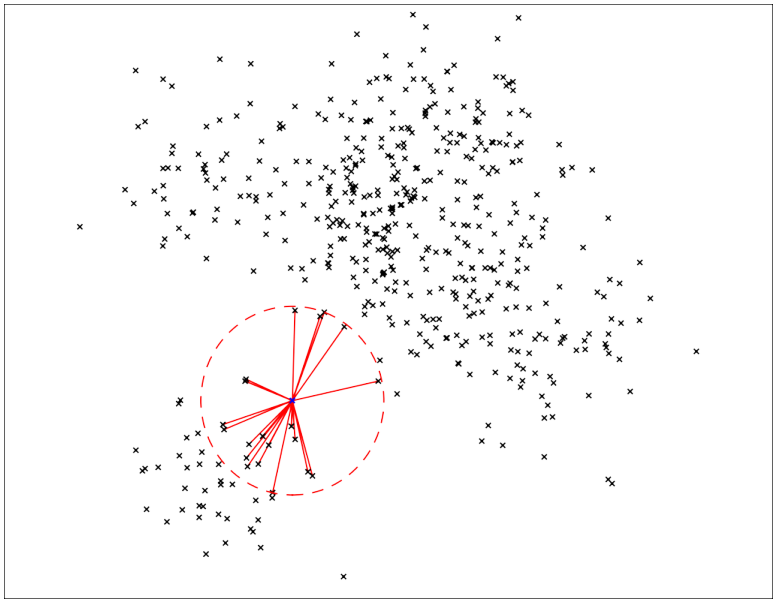


Figure 3.5: K-Nearest Neighbor [7]

3.2.5 Decision Tree Classifier

Another type of supervised machine learning algorithm is a Decision Tree Classifier (DTC) [56]. A tree has many analogies in life and turns out it is influenced by a wide area of machine learning covering both classification and regression trees. Otherwise known as caught. A Decision Tree is a structure where each internal node denotes a test on an attribute each branch represents an outcome of a test and each leaf or terminal node holds a class label. The topmost node in a tree is the root node. In decision analysis, a Decision Tree can be used to individually and explicitly represents decisions and decision making as the name it use a tree, like a model of decisions. So the advantages of Decision Tree is: (i) it is simple to understand interpret and visualize, (ii) Decision Tree implicitly perform variable screening or feature selection, (iii) it can handle both numerical as well as categorical data, (iv) it can also handle multi-output problems decision trees requires relatively little effort from the user for data preparation and (v) nonlinear relationships between parameters do not affect the clip performance. The disadvantage of cost, however, is that the decision tree learns can create over complex trees that do not generalize the data well. This is also known as over-fitting.

3.2.6 Gaussian Naive Bayes

Gaussian Naive Bayes (GNB) [8] is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model".

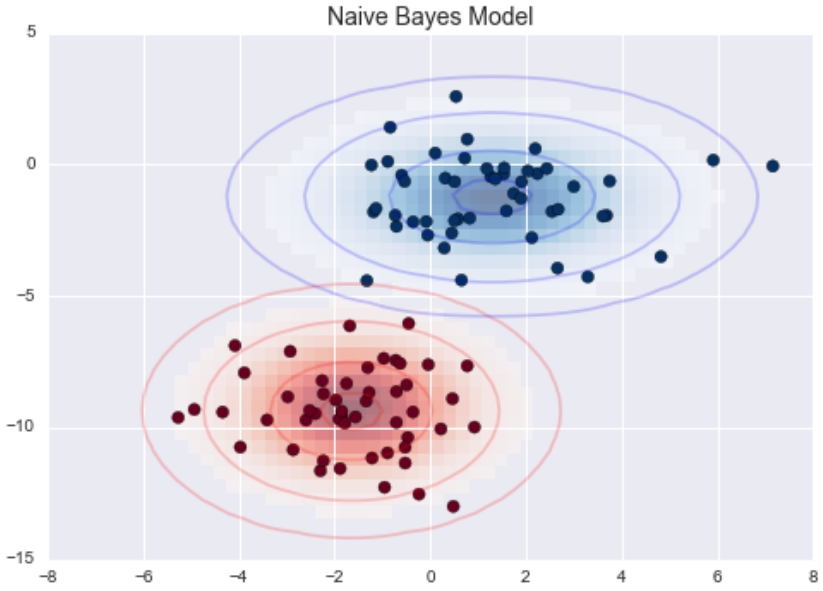


Figure 3.6: Gaussian Naive Bayes [8]

3.3 Performance evaluation

We have used cross-validation sampling technique to tune the parameters of our classification algorithm, feature selection and evaluate the performance of the predictor. Cross-validation [58] is a robust method for sampling instances and widely used in the literature. In k-fold cross-validation, sometimes called rotational estimation, the dataset D is randomly divided into k subsets or folds ($D_1, D_2, ..., D_k$) of approximately equal size. The algorithms are trained and tested k times; each time $t \in 1, 2, ..., k$, it is trained on $D \setminus D_t$ and tested on D_t. The cross-validation estimated accuracy is the overall number of correct classifications, divided by the number of instances in the

dataset. Formally, let $D_{(i)}$ be the test set that includes instance $x_i = \langle v_i, y_i \rangle$, then the cross-validation estimate of accuracy,

$$acc_{cv} = \frac{1}{n} \sum_{\langle v_i, y_i \rangle \in D} \delta(L(D \setminus D_{(i)}, v_i)) \tag{3.7}$$

According to k-fold cross-validation, to train and test our data, we have tried k-fold cross-validation for the six popular machine learning algorithms described in the previous section where $k = 2, 5, 10, 20$. To select appropriate parameters for each classifier, we have used grid search [59, 60] on our sample data using k-fold cross-validation where also $k = 2, 5, 10, 20$. By doing different experiments, we have found the best parameters for each of those methods. In this study, we have tried all those machine learning algorithms from Scikit-learn library [61] in python programming.

In this paper, we mostly have used seven measures to evaluate the performance of the algorithms for σ^{70} promoter prediction. They are: accuracy (Acc), sensitivity (S_n), specificity (S_p), $F1$-Score, area under precision-recall curve (auPR), area under receiver operating characteristic curve (auROC), and Mathew's Correlation Coefficient (MCC). For any two-label classification problem accuracy (Acc) can be represented formally.

$$Acc = \frac{TN + TP}{TP + FN + TN + FP} \tag{3.8}$$

Here, TP represents true positive, that is the total number of correctly classified positive instances. TN represents true negative, that is correctly classified negative instances or actual class type is negative. FP represents false positive means that predicted class type is positive but the actual class type is negative. FN represents false negative which means that the predicted class type is negative but the actual class type is positive. The value of accuracy range between 0% to 100%.

Sensitivity (S_n) or recall is another important metric which is the true positive rate or number of correctly classified positive instances over the total number of positive examples that can be formally defined as follows:

$$S_n = \frac{TP}{TP + FN} \tag{3.9}$$

Similar to specificity (S_p) which is the true negative rate or number of correctly classified negative instances over the total number of negative examples that can be

formally defined as follows:

$$S_p = \frac{TN}{TN + FP} \tag{3.10}$$

F1-score is the harmonic mean of precision and recall. It is defined as:

$$F_1\text{-}Score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3.11}$$

where *precision* is defined by the ratio of the number of true positives over the total number of positive predictions.

$$precision = \frac{TP}{TP + FP} \tag{3.12}$$

All these measures the range of values is in {0,1}. A higher value of these metrics indicates a better performing predictor.

Mathew's correlation coefficient (MCC) is another measure approach to evaluate the model. The range of values is in [-1, 1] which respectively denotes negative classification correlation and positive classification correlation. We can calculate MCC directly from confusion matrix using the following formula,

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3.13}$$

Area under the precision recall curve (auPR) and area under the receiver operating characteristic curve (auROC) are important measures. They reveal the strength of the underlying classifier regardless of the threshold chosen.

Using these measures, it is very important to choose sampling techniques for classification algorithms. Most common are independent test sets and cross-validations. In this paper, we have used cross-validation since they are robust, reduces over-fitting and widely used in the literature of sigma promoters prediction and thus suitable for comparison of different algorithms.

3.4 Summary

In this chapter, we have discussed the dataset and how the dataset can be formulated. To formulate the dataset we have used different approaches such as statistical

measures, k-mer Composition, g-gapped k-mer composition, pattern finding and positioning distance count. We also have considered all those approaches for the different short-range of DNA using the window approach. In parallel, it is also discussed how the features have selected using AdaBoost algorithm. In addition, our proposed algorithms with their performance measure parameters, accuracy (Acc), sensitivity (S_n), specificity (S_p), $F1$-Score, area under precision-recall curve (auPR), area under receiver operating characteristic curve (auROC), and Mathew's Correlation Coefficient (MCC), are discussed in this chapter.

Chapter 4

Experimental Analysis

An experimental analysis is a procedure to agree, refute, or validate a hypothesis. Providing reason and impact insight by displaying what has happened when a specific factor of the exam is disguised. It may happen a considerable change in the goal of the study and scale, but the results are always dependent on the repeated methodology and logical analysis. There also exists natural experimental studies.

4.1 Experimental Results

For experimental purposes, we have implemented all the methods using python programming for data analysis and statistical testing. We also have used the Scikit-learn library [61] where most of the machine learning algorithms are available. In addition, we have plotted Box-plot [62] and Receiver Operating Characteristic (ROC) curve for visually comparing the experimental results.

There were three major experiments done in our work. For the first experiments, we have extracted all our features based on 81 bp full length inputted DNA segments. All the sequences were randomly divided into two sets: one was the training data set and another one was the test data set. The test set was completely independent from the training set in each analysis. The dataset has been divided into k equal parts for k-fold cross-validation. In that case, $k-1$ parts were used for training and the kth part was used for testing. In 10-fold cross-validation, the procedure was continuing 10 times in the manner of rotation. The process is depicted in Figure 4.1. Within the 10-fold cross-validation, we have applied six popular machine learning algorithms. They

were Support Vector Machines (SVM) [6], Logistic Regression (LR) [55], K-Nearest Neighbor (KNN) [7], Decision Tree Classifier (DTC) [56], Gaussian Naive Bayes (GNB) [8] and Linear Discriminant Analysis (LDA) [57]. We have reported the experimental results achieved by these six algorithms in Table 4.1. From the results that has been obtained in the experiments, it is clear that the best performing method was SVM. Logistic regression was also working very nicely on those set of all features and the performances were very close to that of SVM. To illustrate the performance of the algorithms in a better way, present a box-plot of accuracy of different algorithms in Figure 4.2 (a) and ROC curve [63] in Figure 4.2 (b) on the decisions provided by those six classifiers. Please note that box-plot shows a few outliers [64]. After that, we have picked up the best classifier among them based on their different scores such as Acc, auROC, auPR, S_p, S_n, MCC and F_1 Score(Table 4.1). In this case, it was SVM. To see the decision fluctuation, we have plotted the ROC in Figure 4.4 (c) for this best classifier SVM with the changes in its 10-fold cross-validation for each fold.

Secondly, we have applied the same procedure, but in that case, we also have considered the short rage of DNA segments for feature extraction formulated by multiple windowing approaches and also including the full 81 bp length sequence. Window size was tuned for different experiments. Then, we have run those previous six classifiers on these new features group using 10-fold cross-validation. Figure 4.3 depicts the idea of the set of experiments performed at this stage. This time we got a significant improvement in the results as shown in Table 4.2.

From the results reported in Table 4.2, we can note that the best performing classifiers were again Logistic Regression and SVM, where latter was slightly inferior in comparison. Box-plots is shown in Figure 4.4 (a) reveals that there were no outliers after using multiple windowing approaches. We also plotted again ROC for all classifiers in Figure 4.4 (b) and we have found less fluctuation for the best classifier among the folds as shown in Figure 4.4 (c) comparing to the previous set of experiments.

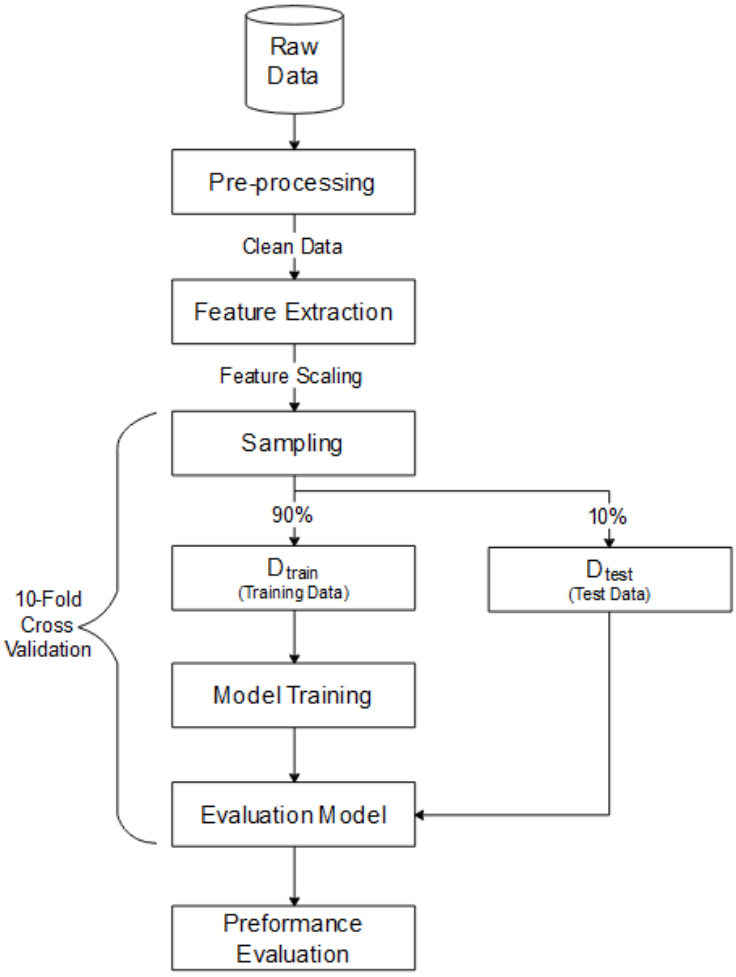
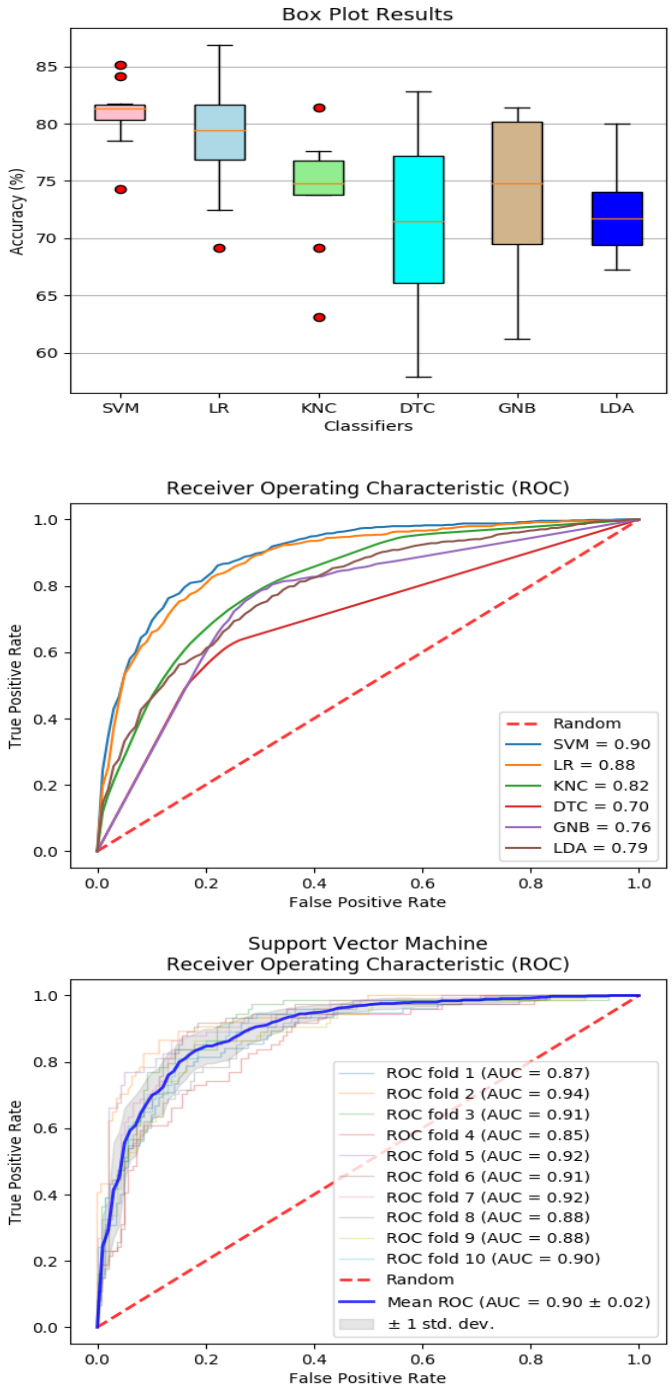


Figure 4.1: Showing typical classification approach

Table 4.1: Experimental Results for the approach in Figure 4.1.

Classifiers	Acc	auROC	auPR	S_p	S_n	MCC	F_1
SVM	**82.81%**	**0.901**	0.829	89.93%	69.37%	0.612	0.735
LR	81.64%	0.882	0.804	81.57%	81.78%	0.614	0.755
KNN	75.90%	0.812	0.646	82.21%	63.97%	0.465	0.648
DTC	71.18%	0.681	0.488	78.07%	58.16%	0.363	0.581
GNB	74.54%	0.764	0.569	74.29%	75.03%	0.474	0.671
LDA	73.52%	0.787	0.668	78.93%	63.29%	0.419	0.623



(a) **(b)** **(c)**

Figure 4.2: Graphical depiction of performance of the method of Figure 4.1: (a) Box-plot of different classifiers on the dataset among cross-folds; (b) Receiver Operating Characteristic curve of different classifiers; (c) Receiver Operating Characteristic curve of different folds in the cross-validation of SVM, the best performing classifier.

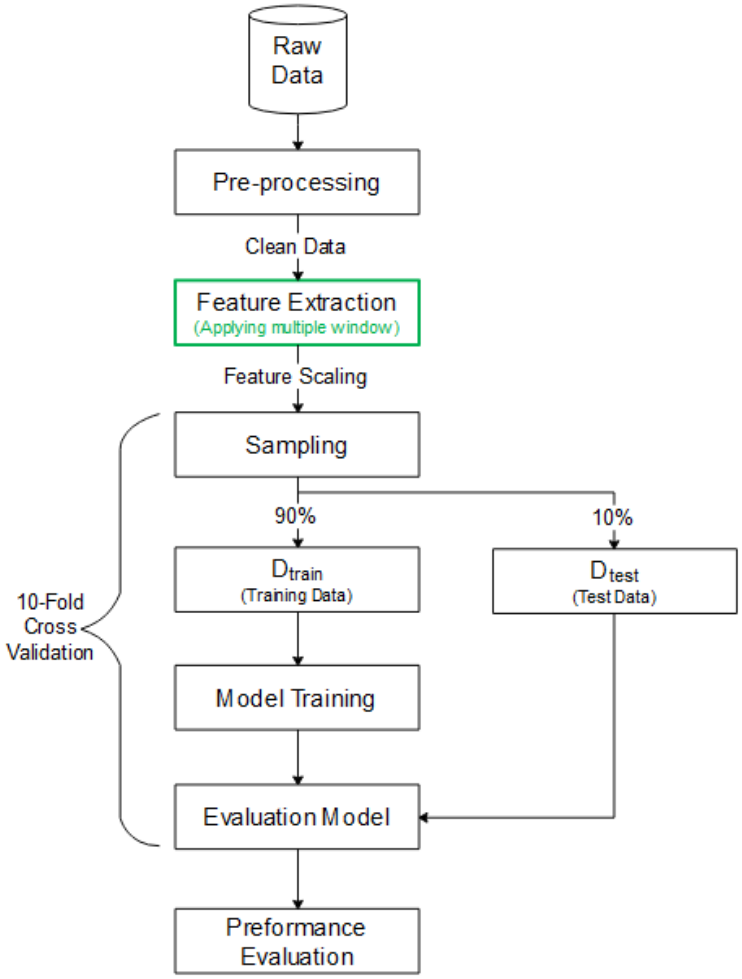
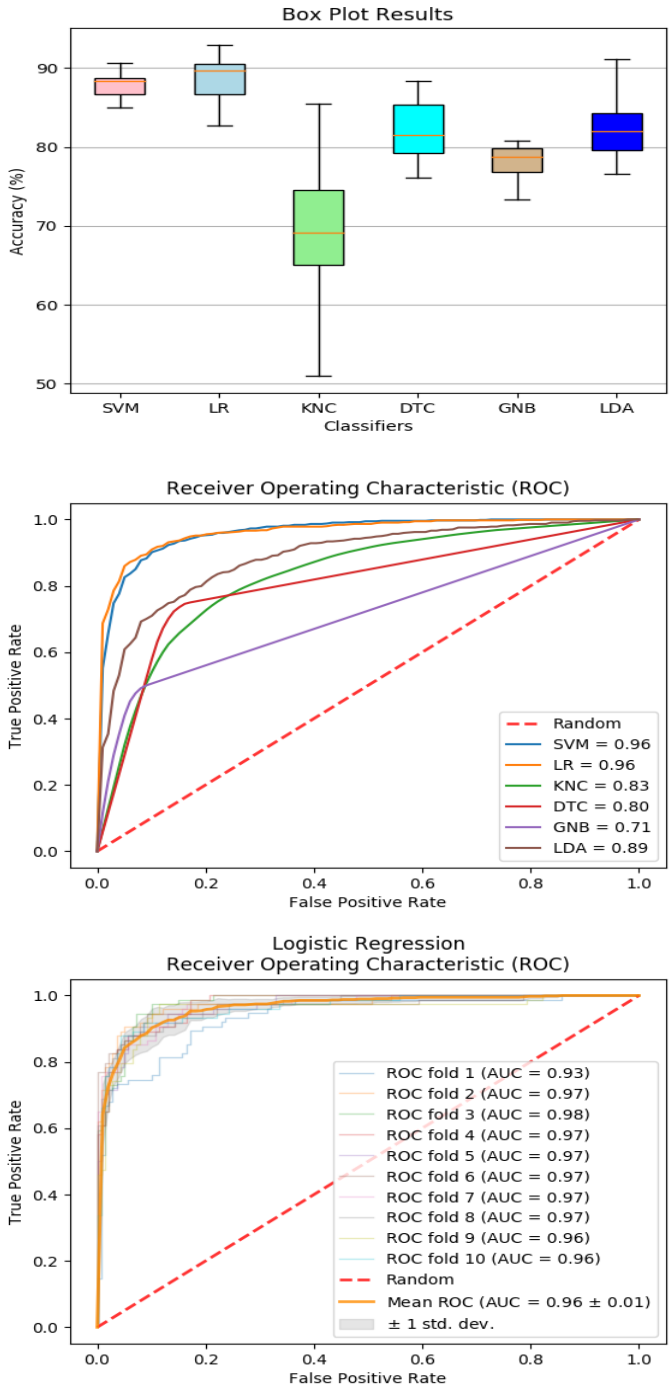


Figure 4.3: Applying window to feature extraction

Table 4.2: Experimental results for the approach in Figure 4.3.

Classifiers	Acc	auROC	auPR	S_p	S_n	MCC	F_1
SVM	88.70%	0.962	0.937	97.07%	72.87%	0.748	0.816
LR	**89.54%**	**0.966**	0.949	87.71%	92.98%	0.783	0.862
KNN	70.95%	0.829	0.670	63.00%	85.96%	0.467	0.672
DTC	82.39%	0.803	0.644	87.00%	73.68%	0.609	0.742
GNB	78.42%	0.713	0.577	94.50%	48.04%	0.504	0.605
LDA	83.05%	0.889	0.831	87.86%	73.95%	0.623	0.752



(a) **(b)** **(c)**

Figure 4.4: Graphical depiction of performance of the method of Figure 4.3: (a) Box-plot of different classifiers on the dataset among cross-folds; (b) Receiver Operating Characteristic curve of different classifiers; (c) Receiver Operating Characteristic curve of different folds in the cross-validation of Logistic Regression, the best performing classifier.

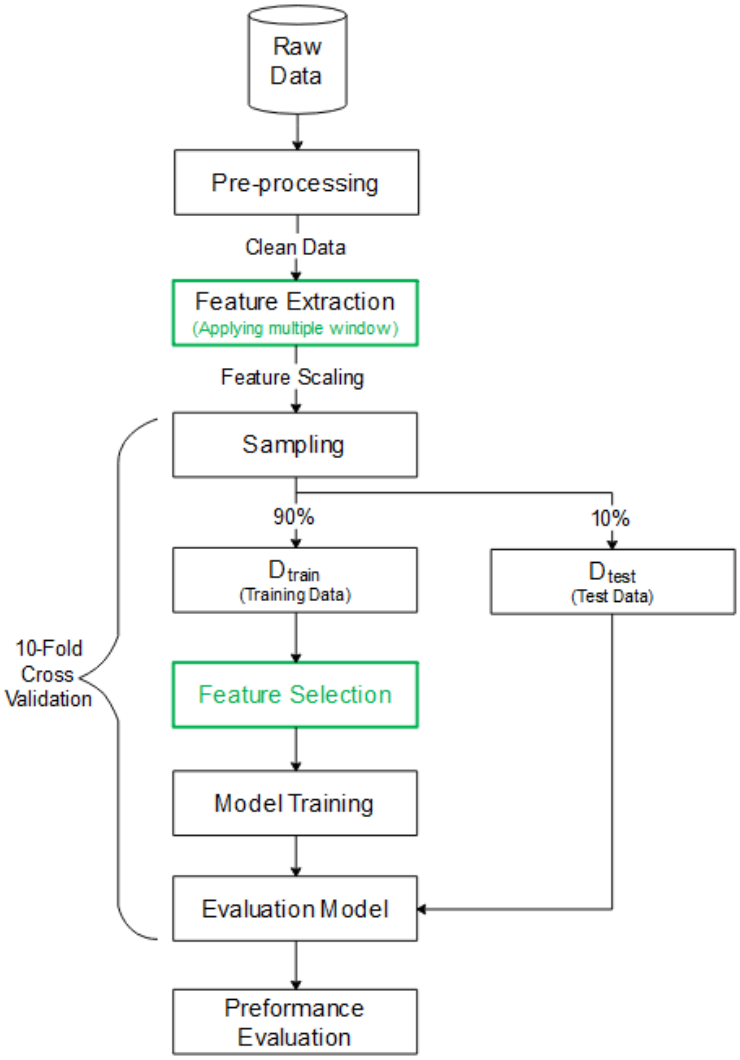
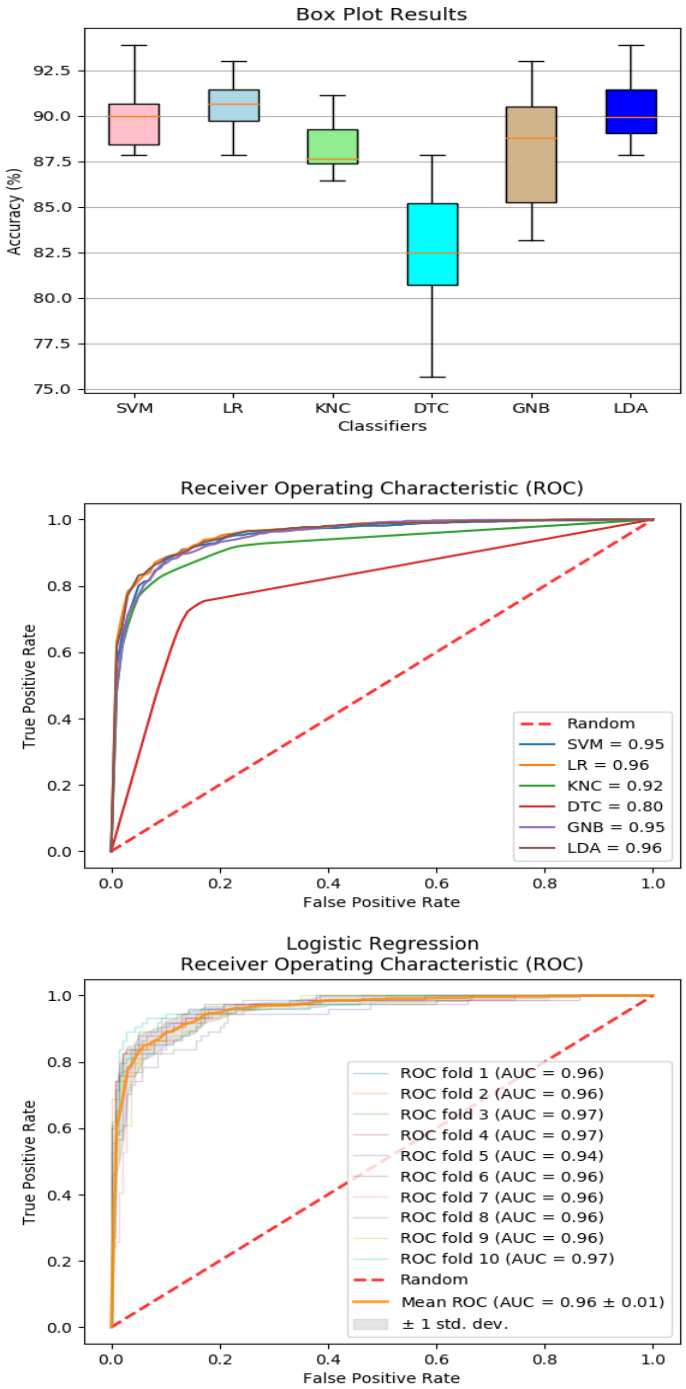


Figure 4.5: After feature selection on both short-range and long-range DNA segments

Table 4.3: Results for the approach in Figure 4.5.

Classifiers	Acc	auROC	auPR	S_p	S_n	MCC	F_1
SVM	89.96%	0.951	0.922	93.57%	83.13%	0.776	0.851
LR	**90.57%**	**0.959**	0.937	94.43%	83.27%	0.789	0.859
KNN	88.32%	0.922	0.870	95.36%	75.03%	0.738	0.816
DTC	82.67%	0.810	0.653	86.50%	75.44%	0.618	0.751
GNB	88.28%	0.952	0.912	88.79%	87.31%	0.748	0.839
LDA	90.24%	0.958	0.938	96.00%	79.35%	0.782	0.849



(a) **(b)** **(c)**

Figure 4.6: Graphical depiction of performance of the method of Figure 4.5: (a) Box-plot of different classifiers on the dataset among cross-folds; (b) Receiver Operating Characteristic curve of different classifiers; (c) Receiver Operating Characteristic curve of different folds in the cross-validation of Logistic Regression, the best performing classifier.

Finally, we have tried to find the minimal set of features in terms of their presence within all 10-fold cross-validation from feature selection by AdaBoost algorithm [49]. The idea of the experiments is depicted in Figure 4.5. Surprisingly, we have noticed that only 27 features are enough to lead most of our classifiers to predict the data with a high accuracy. Figure 4.7 is showing individual feature's accuracy scores for all the classifiers used in this study. All of the classifiers did great except KNN. After that, once more we have trained our classifiers only by those 27 (Table 4.4) features and test the data.

Now, we have seen an increase in results again as shown in Table 4.3. Also note the smooth box-plot and ROC curves seen in Figure 4.6 (a), Figure 4.6 (b) and an acceptable fluctuation in ROC Figure 4.6 (c) for Logistic Regression, the best performing classifier in this case. Therefore, based on the overall accuracy and other measures found among all these experiments, we have selected Logistic Regression as a decision algorithm in our sequence-based predictor named "iPro70-FMWin" since it had the highest accuracy from Logistic Regression.

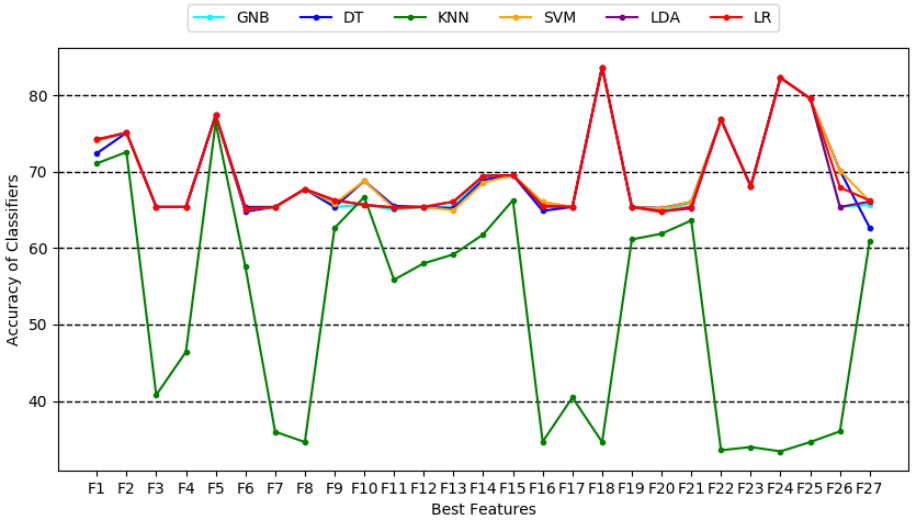


Figure 4.7: The accuracy of best 27 individual features by six popular classifiers

In this work, our feature selection method picked only 27 best features based on their frequency in all 10-folds. For the inquisitiveness, we also examined all classifiers for the other features which had the frequency of less than 10 in 10-folds. All the scores were increased gradually while we were considering the features by decreasing

frequency restriction one by one such as $F = 9, 8, 7, ..., 1$ (Table 4.5). As it can be seen at Table 4.5, the number of features were increased while we considered frequencies by decreasing manner. According to the goal of our feature selection method, we have tried to optimize the features as less as possible. However, we agree that many previous works have done great by using some special pattern of nucleotides such as TATAAT, TTATAA, and TTGACA within specific regions, we also have taken into account those patterns exactly or partially in this work. But it can be seen that from Table 4.4 and Figure 4.7, there were a great influence of g-gap features in term of identifying σ^{70} promoter sequence. According to Figure 4.7, the most effective features in this work was frequency count of "F18" which is TAXXXT (X=A, C, G or T) within -15 bp to -6 bp in Table 4.4. We also have found only "F5" is an important motif among our 27 in Figure 4.7 in a sense where all our classifiers were aligned based on their accuracy, but for the other case, most of their curve were overlapped except KNN. It was happiest to see that most of the features were playing an effective role for all classifiers except KNN.

Table 4.4: Features were found in frequency 10 in 10-fold cross-validation after feature selection.

Feature No.	Feature Index	Pattern	Window Size
4	F1	G + C	-60 bp to +20 bp
24	F2	G + C	-15 bp to -6 bp
105	F3	GC-Skew	-1 bp to -1 bp
107	F4	GC-Skew	1 bp to 1 bp
171	F5	TA	-15 bp to -6 bp
237	F6	TTG	-60 bp to 20 bp
267	F7	CTA	-40 bp to -26 bp
3152	F8	TTGAC	-40 bp to -26 bp
16628	F9	C<– 9 gap –>C	-60 bp to 20 bp
17055	F10	A<– 1 gap –>A	-15 bp to -6 bp
17375	F11	CA<– 4 gap –>A	-60 bp to 20 bp
17458	F12	GA<– 5 gap –>T	-60 bp to 20 bp
17602	F13	TA<– 7 gap –>T	-60 bp to 20 bp
17886	F14	AT<– 12 gap –>T	-60 bp to 20 bp
18510	F15	TT<– 21 gap –>T	-60 bp to 20 bp
18890	F16	TG<– 3 gap –>T	-40 bp to -26 bp
19188	F17	GC<– 8 gap –>C	-40 bp to -26 bp
19586	F18	TA<– 3 gap –>T	-15 bp to -6 bp
19993	F19	A<– 3 gap –>GG	-60 bp to 20 bp
20492	F20	T<– 10 gap –>TC	-60 bp to 20 bp
20826	F21	A<– 16 gap –>GT	-60 bp to 20 bp
22162	F22	A<– 2 gap –>AT	-15 bp to -6 bp
22166	F23	A<– 2 gap –>CT	-15 bp to -6 bp
22543	F24	TATAAT	-15 bp to -6 bp
22563	F25	TTATAA	-15 bp to -6 bp
22567	F26	TTGACA	-40 bp to -26 bp
22594	F27	DNaseI	-15 bp to -6 bp

Table 4.5: Feature selection with their frequency within 10-folds and result of the best classifier for the corresponding features.

No. of Features	Freq	Best classifier	Acc	auROC	auPR	S_p	S_n	MCC	F_1
27	10	LR	90.57	0.959	0.937	94.43	83.27	0.789	0.859
55	9	LR	91.78	0.973	0.955	94.71	86.23	0.817	0.879
83	8	SVM	93.13	0.979	0.966	95.07	89.47	0.848	0.900
112	7	SVM	93.69	0.982	0.972	96.00	89.34	0.859	0.907
156	6	SVM	94.35	0.984	0.974	95.43	92.31	0.876	0.919
216	5	SVM	95.89	0.992	0.987	97.36	93.12	0.909	0.940
331	4	SVM	96.36	0.994	0.990	97.71	93.79	0.919	0.947
488	3	SVM	96.36	0.995	0.991	98.07	93.12	0.919	0.946
816	2	SVM	96.82	0.996	0.993	98.36	93.93	0.930	0.953
1731	1	SVM	96.92	0.995	0.991	98.50	93.93	0.932	0.955

4.2 Comparison with Previous Methods

A large number of features have been used in many previous methods [11, 12, 65] to recognize promoter sequence. While the goal of this research was to optimize the number of features into a small number of effective features. However, we have compared the performance of our algorithm with two other state-of-the-art algorithms. The experimental report of iPro70-FMWin was recorded in contrast with three other popular methods: iPro70-PseZNC [12], IPDM [66] and Z-curve [67] in Table 4.6.

Table 4.6: Performance comparison of iPro70-FMWin with other different method

Method	Acc	auROC	auPR	S_p	S_n	MCC	F_1
IPDM	89.2%	0.953	0.920	91.4%	84.9%	0.761	—
iPro70-PseZNC	84.5%	0.909	—	86.8%	80.3%	0.663	—
Z-curve	77.8%	0.848	—	79.5%	74.6%	0.527	—
iPro70-FMWin	**90.57%**	**0.959**	**0.937**	**94.43%**	**83.27%**	**0.789**	**0.859**

4.3 Web-server Implementation

We have designed our feature extraction model in python programming only for those 27 effective features. We also have implemented Logistic Regression for classification in our sequence-based predictor named "iPro70-FMWin" for identifying σ^{70} promoter in the prokaryote. For the benefit of researchers, a user-friendly online service was built and can be freely accessible at `http://ipro70.pythonanywhere.com/server`. The web-server interface is look like as Figure 4.8.

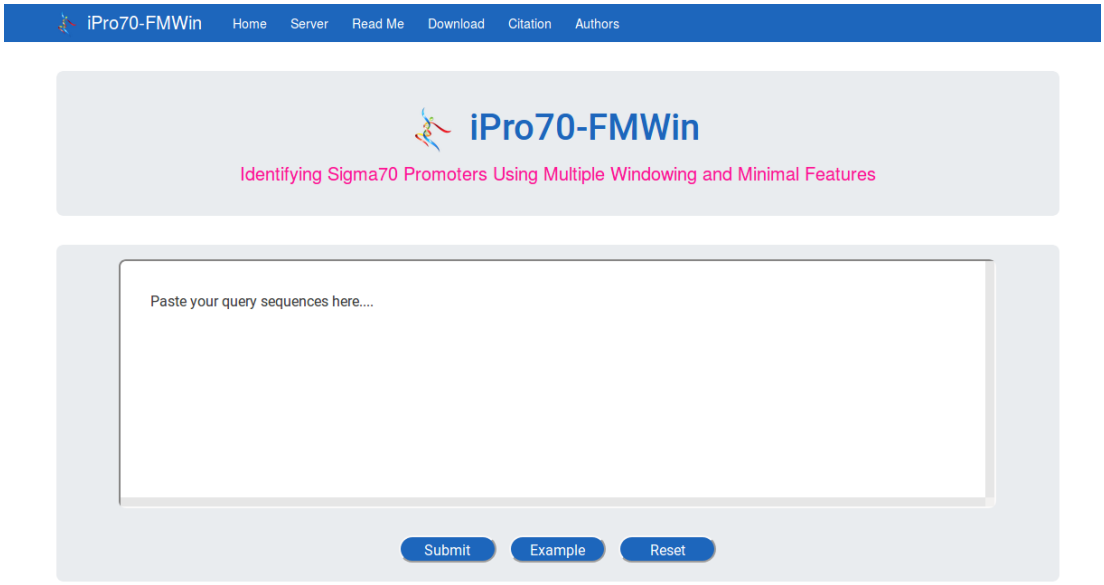


Figure 4.8: Web-server for practical testing

Now, if anyone wants to test their DNA sequence whether it is promoter or not, users should put their DNA sequence in input text box (see Figure 4.9). Then, after clicking the submit button the server will response instantly with the predictive scores.

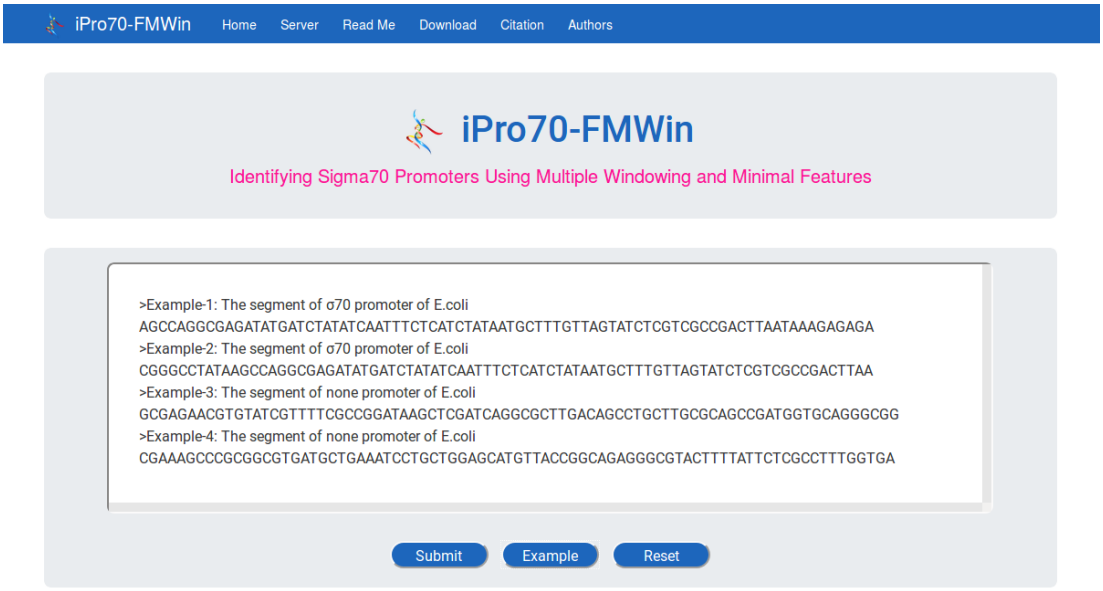


Figure 4.9: Web-server with sample input

The green colors are showing the results are positive which means that the inputted sequences are promoter sequence. The red colors are showing the results are negative sequence which means that the inputted sequences are are non-promoter sequences (see Figure 4.10).

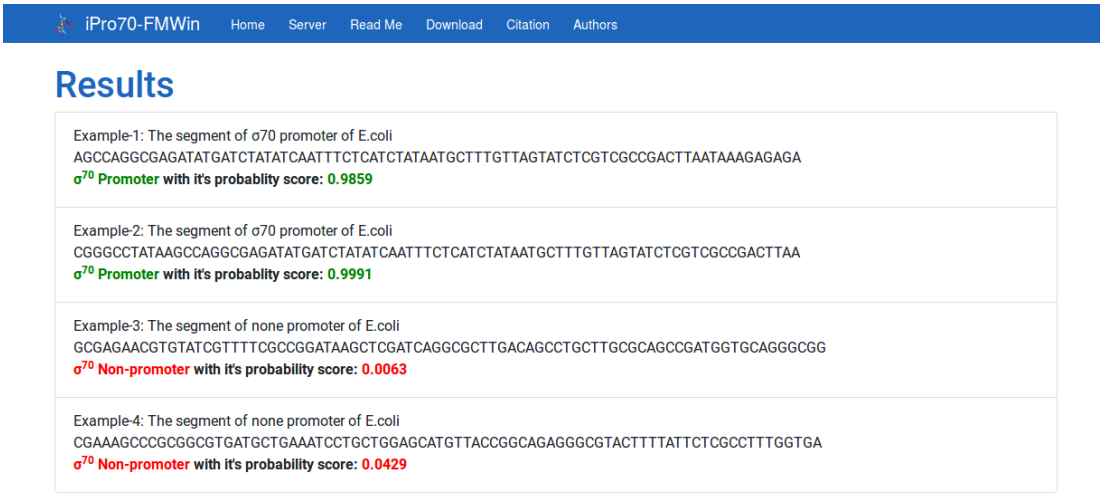


Figure 4.10: Web-server with results for sample input

4.4 Summary

In this chapter, we have shown mainly three approaches with their experimental results. First, we have implements our proposed machine learning classifiers in python using Scikit-learn package. Then, we have generated features using our typical feature extraction methods. After we have trained and tested all those classifiers for the performance evolution. Again, we tried same procedure but in this case, we have also considered about the features within the short-range of DNA segments using window approach. Finally, we have selected a small number of features which are most influential to leads our classifiers for identifying σ^{70} promoter as well as we have compared our method with two other state-of-the-art algorithms. For all the procedures, we have plotted the different graph for visualization. At the end of this chapter, we have discussed our web-server that we have implemented for practical testing.

Chapter 5

Conclusion

In this thesis, we have proposed a prediction method iPro70-FMWin. We have used an optimal number of features extracted from sequences only. We believe that our features are more effective to identify σ^{70} promoter sequences. In addition, we also believe that our 27 features may be useful to identify the other promoter sequences. In parallel, our method did better in terms of performance compared two other state-of-the-art algorithms.

Although the thesis work was able to identify sigma 70 promoter with a high performance, but the problem was in the small number of dataset. Another, the dataset was not balance. Therefore, we were not able to apply some other effective methods. For instance, deep neural network is working great in the area of bioinformatics. But for deep neural network dataset should be large.

The future work will focus on different promoter sequences and comparison study among them. Moreover, this role can play a significant supplementary in the other marginalized way for the prediction of promoters and transcription start sites. Besides, there is still a great place to improve the accuracy of prediction.

Bibliography

[1] Wikipedia, "Protein biosynthesis," 2018, [Online; accessed 02-August-2018]. [Online]. Available: https://en.wikipedia.org/wiki/Protein_biosynthesis x, 2

[2] W. Commons, "File:pdb 2h27 ebi.jpg — wikimedia commons, the free media repository," 2009, [Online; accessed 24-April-2018]. [Online]. Available: https://commons.wikimedia.org/w/index.php?title=File:PDB_2h27_EBI.jpg&oldid=18274510 x, 3

[3] Matthew-Hunter, "The sigma factor is involved in the initiation process of prokaryotic transcritption," 2009, [Online; accessed 01-August-2018]. [Online]. Available: http://helicase.pbworks.com/w/page/17605688/Matthew-Hunter x, 4

[4] CLIPARTUSE, "Image from clipartuse.com," 2018, [Online; accessed 01-August-2018]. [Online]. Available: https://clipartuse.com/clipart/1329132 x, 6

[5] kathleenhalme, "Science lab clipart black and white," 2018, [Online; accessed 01-August-2018]. [Online]. Available: https://kathleenhalme.com x, 6

[6] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995. x, 19, 20, 28

[7] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992. x, 19, 21, 22, 28

[8] K. P. Murphy, "Naive bayes classifiers," *University of British Columbia*, vol. 18, 2006. x, 19, 23, 28

[9] T. M. Gruber and C. A. Gross, "Multiple sigma subunits and the partitioning of bacterial transcription space," *Annual Reviews in Microbiology*, vol. 57, no. 1, pp. 441–466, 2003. 2

[10] M. Towsey, P. Timms, J. Hogan, and S. A. Mathews, "The cross-species prediction of bacterial promoters using a support vector machine," *Computational biology and chemistry*, vol. 32, no. 5, pp. 359–366, 2008. 4

[11] 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, pp. 12 961–12 972, 2014. 5, 9, 11, 14, 38

[12] H. Lin, Z. Liang, H. Tang, and W. Chen, "Identifying sigma70 promoters with novel pseudo nucleotide composition." *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 10, 2017. 5, 11, 13, 14, 16, 38

[13] B. Demeler and G. Zhou, "Neural network optimization for e. coli promoter prediction," *Nucleic acids research*, vol. 19, no. 7, pp. 1593–1599, 1991. 5, 9, 10

[14] A. Lukashin, V. Anshelevich, B. Amirikyan, A. Gragerov, and M. Frank-Kamenetskii, "Neural network models for promoter recognition," *Journal of Biomolecular Structure and Dynamics*, vol. 6, no. 6, pp. 1123–1133, 1989.

[15] S. d. A. e Silva, F. Forte, I. T. Sartor, T. Andrighetti, G. J. Gerhardt, A. P. L. Delamare, and S. Echeverrigaray, "Dna duplex stability as discriminative characteristic for escherichia coli σ54-and σ28-dependent promoter sequences," *Biologicals*, vol. 42, no. 1, pp. 22–28, 2014. 5

[16] S. Audic and J.-M. Claverie, "Detection of eukaryotic promoters using markov transition matrices," *Computers & chemistry*, vol. 21, no. 4, pp. 223–227, 1997. 5

[17] R. R. Mallios, D. M. Ojcius, and D. H. Ardell, "An iterative strategy combining biophysical criteria and duration hidden markov models for structural predictions of chlamydia trachomatis σ 66 promoters," *BMC bioinformatics*, vol. 10, no. 1, p. 271, 2009. 5

[18] B. Liu, F. Yang, D.-S. Huang, and K.-C. Chou, "ipromoter-2l: a two-layer predictor for identifying promoters and their types by multi-window-based pseknc," *Bioinformatics*, vol. 34, no. 1, pp. 33–40, 2017. 5, 9, 11

[19] J. W. Fickett and A. G. Hatzigeorgiou, "Eukaryotic promoter recognition," *Genome research*, vol. 7, no. 9, pp. 861–878, 1997. 5

[20] B. Grech, S. Maetschke, S. Mathews, and P. Timms, "Genome-wide analysis of chlamydiae for promoters that phylogenetically footprint," *Research in microbiology*, vol. 158, no. 8-9, pp. 685–693, 2007. 5

[21] L. Gordon, A. Y. Chervonenkis, A. J. Gammerman, I. A. Shahmuradov, and V. V. Solovyev, "Sequence alignment kernel for recognition of promoter regions," *Bioinformatics*, vol. 19, no. 15, pp. 1964–1971, 2003. 9, 10

[22] Q.-Z. Li and H. Lin, "The recognition and prediction of σ70 promoters in escherichia coli k-12," *Journal of theoretical biology*, vol. 242, no. 1, pp. 135–141, 2006. 9, 10

[23] F. Griffith, "The significance of pneumococcal types," *Epidemiology & Infection*, vol. 27, no. 2, pp. 113–159, 1928. 9

[24] O. T. Avery, C. M. MacLeod, and M. McCarty, "Studies on the chemical nature of the substance inducing transformation of pneumococcal types: induction of transformation by a desoxyribonucleic acid fraction isolated from pneumococcus type iii," *Journal of experimental medicine*, vol. 79, no. 2, pp. 137–158, 1944. 9

[25] J. S. Fruton, *Proteins, enzymes, genes: the interplay of chemistry and biology*. Yale University Press, 1999. 9

[26] R. Dahm, "Friedrich miescher and the discovery of dna," *Developmental biology*, vol. 278, no. 2, pp. 274–288, 2005. 9

[27] W. Saenger, "Principles of nucleic acid structure modified nucleosides and nucleotides; nucleoside di-and triphosphates; coenzymes and antibiotics," *Principles of Nucleic Acid Structure*, 1984. 10

[28] T. V. Chalikian, J. Völker, G. E. Plum, and K. J. Breslauer, "A more unified picture for the thermodynamics of nucleic acid duplex melting: a characterization by calorimetric and volumetric techniques," *Proceedings of the National Academy of Sciences*, vol. 96, no. 14, pp. 7853–7858, 1999. 10

[29] Z.-Y. Liang, H.-Y. Lai, H. Yang, C.-J. Zhang, H. Yang, H.-H. Wei, X.-X. Chen, Y.-W. Zhao, Z.-D. Su, W.-C. Li *et al.*, "Pro54db: a database for experimentally verified sigma-54 promoters," *Bioinformatics*, vol. 33, no. 3, pp. 467–469, 2017. 11

[30] K.-C. Chou, "Some remarks on protein attribute prediction and pseudo amino acid composition," *Journal of theoretical biology*, vol. 273, no. 1, pp. 236–247, 2011. 12

[31] ——, "Some remarks on predicting multi-label attributes in molecular biosystems," *Molecular Biosystems*, vol. 9, no. 6, pp. 1092–1100, 2013. 12

[32] S. Gama-Castro, H. Salgado, A. Santos-Zavaleta, D. Ledezma-Tejeida, L. Muñiz-Rascado, J. S. García-Sotelo, K. Alquicira-Hernández, I. Martínez-Flores, L. Pannier, J. A. Castro-Mondragón *et al.*, "RegulonDB version 9.0: high-level integration of gene regulation, coexpression, motif clustering and beyond," *Nucleic acids research*, vol. 44, no. D1, pp. D133–D143, 2015. 13

[33] Y. Gan, J. Guan, and S. Zhou, "A comparison study on feature selection of dna structural properties for promoter prediction," *BMC bioinformatics*, vol. 13, no. 1, p. 4, 2012. 14

[34] J. Shin and V. Noireaux, "Efficient cell-free expression with the endogenous e. coli rna polymerase and sigma factor 70," *Journal of biological engineering*, vol. 4, no. 1, p. 8, 2010. 14

[35] H. Yamagishi, "Nucleotide distribution in bacterial dna's differing in g+ c content," *Journal of molecular evolution*, vol. 3, no. 3, pp. 239–242, 1974. 14

[36] J. Lobry, "Asymmetric substitution patterns in the two dna strands of bacteria." *Molecular biology and evolution*, vol. 13, no. 5, pp. 660–665, 1996. 14

[37] P. A. Ginno, Y. W. Lim, P. L. Lott, I. Korf, and F. Chédin, "Gc skew at the 5 and 3 ends of human genes links r-loop formation to epigenetic regulation and

transcription termination," *Genome research*, vol. 23, no. 10, pp. 1590–1600, 2013. 14

[38] P. E. Compeau, P. A. Pevzner, and G. Tesler, "How to apply de bruijn graphs to genome assembly," *Nature biotechnology*, vol. 29, no. 11, p. 987, 2011. 14

[39] K. E. Samocha, E. B. Robinson, S. J. Sanders, C. Stevens, A. Sabo, L. M. McGrath, J. A. Kosmicki, K. Rehnström, S. Mallick, A. Kirby *et al.*, "A framework for the interpretation of de novo mutation in human disease," *Nature genetics*, vol. 46, no. 9, p. 944, 2014.

[40] V. Aggarwala and B. F. Voight, "An expanded sequence context model broadly explains variability in polymorphism levels across the human genome," *Nature genetics*, vol. 47, no. 3, p. 349, 2015. 14

[41] A. M. Huerta and J. Collado-Vides, "Sigma70 promoters in escherichia coli: specific transcription in dense regions of overlapping promoter-like signals," *Journal of molecular biology*, vol. 333, no. 2, pp. 261–278, 2003. 15

[42] D. G. Olson, M. Maloney, A. A. Lanahan, S. Hon, L. J. Hauser, and L. R. Lynd, "Identifying promoters for gene expression in clostridium thermocellum," *Metabolic Engineering Communications*, vol. 2, pp. 23–29, 2015.

[43] G. D. Stormo, "Dna binding sites: representation and discovery," *Bioinformatics*, vol. 16, no. 1, pp. 16–23, 2000. 15

[44] M. El Hassan and C. Calladine, "Propeller-twisting of base-pairs and the conformational mobility of dinucleotide steps in dna," *Journal of molecular biology*, vol. 259, no. 1, pp. 95–103, 1996. 16

[45] G. E. Crawford, I. E. Holt, J. Whittle, B. D. Webb, D. Tai, S. Davis, E. H. Margulies, Y. Chen, J. A. Bernat, D. Ginsburg *et al.*, "Genome-wide mapping of dnase hypersensitive sites using massively parallel signature sequencing (mpss)," *Genome research*, vol. 16, no. 1, pp. 123–131, 2006.

[46] A. P. Boyle, S. Davis, H. P. Shulha, P. Meltzer, E. H. Margulies, Z. Weng, T. S. Furey, and G. E. Crawford, "High-resolution mapping and characterization of open chromatin across the genome," *Cell*, vol. 132, no. 2, pp. 311–322, 2008. 16

[47] M. Dash and H. Liu, "Feature selection for classification," *Intelligent data analysis*, vol. 1, no. 3, pp. 131–156, 1997. 17

[48] K. Kira and L. A. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in *Aaai*, vol. 2, 1992, pp. 129–134. 18

[49] L. Shen and L. Bai, "Adaboost gabor feature selection for classification," in *Proc. of Image and Vision Computing NewZealand*, 2004, pp. 77–83. 18, 19, 35

[50] Y.-W. Chen and C.-J. Lin, "Combining svms with various feature selection strategies," in *Feature extraction.* Springer, 2006, pp. 315–324. 18

[51] R. Díaz-Uriarte and S. A. De Andres, "Gene selection and classification of microarray data using random forest," *BMC bioinformatics*, vol. 7, no. 1, p. 3, 2006. 18

[52] Q. Cheng, P. K. Varshney, and M. K. Arora, "Logistic regression for feature selection and soft classification of remote sensing data," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 4, pp. 491–494, 2006. 18

[53] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning.* Springer, 2013, vol. 112. 18

[54] M. L. Bermingham, R. Pong-Wong, A. Spiliopoulou, C. Hayward, I. Rudan, H. Campbell, A. F. Wright, J. F. Wilson, F. Agakov, P. Navarro *et al.*, "Application of high-dimensional feature selection: evaluation for genomic prediction in man," *Scientific reports*, vol. 5, p. 10312, 2015. 18

[55] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression.* John Wiley & Sons, 2013, vol. 398. 19, 20, 28

[56] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991. 19, 22, 28

[57] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers, "Fisher discriminant analysis with kernels," in *Neural networks for signal processing IX, 1999. Proceedings of the 1999 IEEE signal processing society workshop.* Ieee, 1999, pp. 41–48. 19, 20, 28

[58] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, no. 2. Montreal, Canada, 1995, pp. 1137–1145. 23

[59] C.-W. Hsu, C.-C. Chang, C.-J. Lin *et al.*, "A practical guide to support vector classification," 2003. 24

[60] K. Coussement and D. Van den Poel, "Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques," *Expert systems with applications*, vol. 34, no. 1, pp. 313–327, 2008. 24

[61] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011. 24, 27

[62] P. Kampstra *et al.*, "Beanplot: A boxplot alternative for visual comparison of distributions," 2008. 27

[63] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982. 28

[64] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial intelligence review*, vol. 22, no. 2, pp. 85–126, 2004. 28

[65] J. J. Gordon, M. W. Towsey, J. M. Hogan, S. A. Mathews, and P. Timms, "Improved prediction of bacterial transcription start sites," *Bioinformatics*, vol. 22, no. 2, pp. 142–148, 2005. 38

[66] H. Lin and Q.-Z. Li, "Eukaryotic and prokaryotic promoter prediction using hybrid approach," *Theory in Biosciences*, vol. 130, no. 2, pp. 91–100, 2011. 38

[67] K. Song, "Recognition of prokaryotic promoters based on a novel variable-window z-curve method," *Nucleic acids research*, vol. 40, no. 3, pp. 963–971, 2011. 38