MACHINE LEARNING METHODS FOR PROMOTER REGION PREDICTION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HİLAL ARSLAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JUNE 2011

Approval of the thesis:

MACHINE LEARNING METHODS FOR PROMOTER REGION PREDICTION

submitted by **HİLAL ARSLAN** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı _____
Head of Department, **Computer Engineering**

Assoc. Prof. Dr. Tolga Can _____
Supervisor, **Computer Engineering Department, METU**

Examining Committee Members:

Assist. Prof. Dr. Sinan Kalkan
Computer Engineering Department, METU _____

Assoc. Prof. Dr. Tolga Can
Computer Engineering Department, METU _____

Assist. Prof. Dr. Yeşim Aydın Son
Informatics Institute, METU _____

Assist. Prof. Dr. Ahmet Oğuz Akyüz
Computer Engineering Department, METU _____

Assist. Prof. Dr. Vilda Purutçuoğlu
Statistics Department, METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: HİLAL ARSLAN

Signature :

ABSTRACT

MACHINE LEARNING METHODS FOR PROMOTER REGION PREDICTION

Arslan, Hilal
M.Sc., Department of Computer Engineering
Supervisor : Assoc. Prof. Dr. Tolga Can

June 2011, 61 pages

Promoter classification is the task of separating promoter from non promoter sequences. Determining promoter regions where the transcription initiation takes place is important for several reasons such as improving genome annotation and defining transcription start sites.

In this study, various promoter prediction methods called ProK-means, ProSVM, and 3S1C are proposed. In ProSVM and ProK-means algorithms, structural features of DNA sequences are used to distinguish promoters from non promoters. Obtained results are compared with ProSOM which is an existing promoter prediction method. It is shown that ProSVM is able to achieve greater recall rate compared to ProSOM results.

Another promoter prediction methods proposed in this study is 3S1C. The difference of the proposed technique from existing methods is using signal, similarity, structure, and context features of DNA sequences in an integrated way and a hierarchical manner. In addition to current methods related to promoter classification, the similarity feature, which compares the promoter regions between human and other species, is added to the proposed system. We show that the similarity feature improves the accuracy. To classify core promoter regions, firstly, signal, similarity, structure, and context features are extracted and then, these features are classified separately by using Support Vector Machines. Finally, output predictions are

combined using multilayer perceptron. The result of 3S1C algorithm is very promising.

Keywords: promoter prediction, signal features, similarity features, structure features, context features, support vector machines, multilayer perceptron, classification

ÖZ

MAKİNE ÖĞRENİMİ YÖNTEMLERİ İLE PROMOTER BÖLGESİ TAHMİNİ

Arslan, Hilal
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Tolga Can

Haziran 2011, 61 sayfa

Promoter sekanslarını, promoter olmayan sekanslardan ayırma işlemi promoter sınıflandırma olarak adlandırılır. Promoter bölgeleri transkripsiyon başlangıç bölgelerinde yer alıp, bu bölgeleri tanımlamak, gen bilgilerini geliştirme ve transkripsiyon başlangıç bölgelerini tanımlama gibi bir çok sebepten ötürü önemlidir.

Bu çalışmada promoter bölgelerini sınıflandırmak için çeşitli yöntemler önerilmiştir. Bu metotlar ProK-means, ProSVM ve 3S1C olarak adlandırılır. ProSVM ve ProK-means algoritmalarında, promoter bölgelerini tanımlamak için sadece DNA'nın yapısal özellikleri kullanılmıştır. Elde edilen sonuçlar, diğer bir promoter tahmin yöntemi olan ProSOM ile karşılaştırılmıştır. ProSVM yönteminin, ProSOM yönteminden daha iyi sonuç verdiği gösterilmiştir.

Daha sonra, 3S1C yöntemi tanıtılmıştır. Önerilen diğer promoter tahmin yöntemlerinden farkı, burada DNA'nın sinyal, içerik, yapı ve benzerlik özniteliklerinin hepsinin birarada kullanılmasıdır. Ayrıca, promoter sınıflandırma yapan mevcut yöntemlere ek olarak bu çalışmada benzerlik özniteliği tanıtılmıştır. Benzerlik özniteliği, insan ve diğer türler arasındaki promoter bölgelerini karşılaştırır. Ayrıca benzerlik özniteliği, hata payını bir miktar azaltır. Promoter bölgelerini sınıflandırmak için ilk olarak sinyal, içerik, benzerlik ve yapı öznitelikleri çıkartılır. Sonra, bu öznitelikler destek vektör makinaları kullanılarak ayrı ayrı sınıflandırılır.

Son adımda, çok katmanlı sinir ağları kullanarak sonuçlar birleştirilirip, sınıflandırma işlemi tamamlanır. 3S1C algoritma sonucunun umut verici olduğu görülmüştür.

Anahtar Kelimeler: promoter tahmini, sinyal özniteliği, benzerlik özniteliği, yapı özniteliği, içerik özniteliği, destek vektör makinaları, çok katmanlı sinir ağları,sınıflandırma

to my husband Güray Arslan :)

ACKNOWLEDGMENTS

This study was carried out under the supervision of Assoc. Prof. Dr. Tolga Can. I would like to express great appreciation to him for his great support, reviews, advice and patience during this study.

I would also thank Assist. Prof. Dr. Sinan Kalkan, Assist. Prof. Dr. Ahmet Oğuz Akyüz, Yeşim Aydın Son, and Assist. Prof. Dr. Vilda Purutçuoğlu for kindly agreeing to be in my thesis committee.

I would like to thank my unofficial supervisor Güray Arslan for always being with me and his endless support and love. Without his great reviews, there was no way for completing this thesis on time. I would like to thank my family for their invaluable support during this long period.

I am grateful to my friends Gamze Toybıyık, Selma Süloğlu, Mine Yoldaş, Hande Çelikkanat, Aslı Gençtav, and Serdar Çiftci for their nice friendship.

Finally, I would like to thank everyone who believe in me.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Promoter classification is a challenging problem in bioinformatics field due to its complexity. If annotation of all the genes was known and locations of the promoters were at the same place respective to the gene, all promoter regions would be identified correctly. However, there are several uncertainties in gene annotation about the location of promoter regions. Moreover, promoter regions are located around Transcription Start Sites (TSS) and there maybe a lot of TSSs in a gene. In addition, sequence signals related to promoters are not determined by a rule. That is, separating short promoter from non promoter sequences is a complex problem in bioinformatics.

1.1 Background

Cells in human body include a nucleus where the DNA is located. The shape of the DNA is a double helix as illustrated in Figure 1.1.

The complex code of genetic information of human body is contained within the DNA molecule. The information in DNA is stored by four chemical bases: adenine (A), guanine (G), cytosine (C), and thymine (T). They are called as base pairs (bps). Dinucleotide is a sequence of two base pairs, and trinucleotide is a triplet of base pairs. The order of these bases determines genetic information and this order includes instructions to produce a protein which is called a gene. A gene might contain between 1 thousand and 1 million bases.

The most important task of a DNA sequence is producing proteins. Proteins are constituted by smaller molecules called as amino acids. The production of proteins performed by DNA includes two steps:

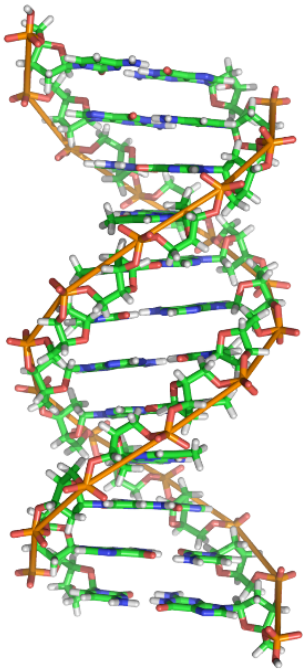

Figure 1.1: Structure of DNA (Adapted from [1])

- Transcription is a process of generating an RNA molecule from a DNA template.

- Generated RNA produces a specific protein and this process is called as translation.

In transcription, some proteins are necessary as enzymes and these proteins bind to a DNA sequence. These specific proteins are generally called as transcription factors.

1.1.1 Genes

A gene is a region of DNA responsible from hereditary characteristics of living organisms. It contains the codes for proteins or RNA chains which have some functions in the organisms. Genome sequences can be divided into two parts: non coding and coding. A non coding sequence is a part of DNA sequence and this part does not encode for the production of proteins. The coding sequences define what the gene produces.

1.1.2 Promoters

A promoter is a region located around Transcription Starts Sites (TSS). A schematic representation of the location of the promoter regions with other regions is illustrated in Figure 1.2. Two nucleotides on opposite complementary DNA or RNA strands which are connected via hydrogen bonds are called a base pair and abbreviated as bp. The promoter region consists of three main parts:

- the core promoter, which includes the binding site for RNA polymerase and which is the minimum part of the promoter needed to start gene transcription. Moreover, this part is generally located about 35 bps upstream of the transcription start site.
- the proximal promoter, which includes several organizer elements and which is located about 250 bp upstream of the transcription start site.
- the distal promoter, which can include additional organizer elements called enhancers and silencers [2]. This part is several thousands of base pairs upstream of the transcription start site.

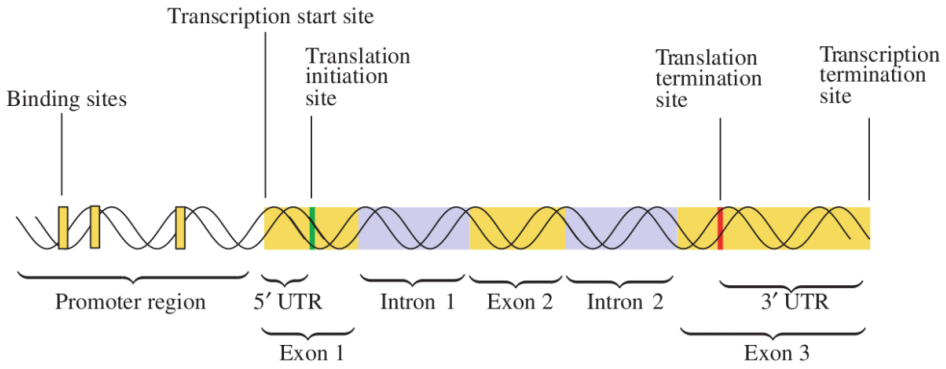

Figure 1.2: Schematic representation of a DNA sequence (Adapted from [3])

Determining the promoter regions is an important task. If the promoter regions are identified correctly, this improves genome annotation which gives a lot of information about the functions in a genome. There are many signals that help us detect promoters, such as TATA boxes and CpG islands. These signals may be located in the promoter region of DNA sequences. TATA box has a consensus sequence TATAAAA, but large deviations from this consensus

have been found in different genes [4]. In CpG island, there are a lot of C and G in the sequence. However, these signals do not guarantee the presence of the promoters. Similarly, their absence does not imply that there is no promoter.

1.2 Contributions of the Thesis

Firstly, we introduced ProK-means and ProSVM algorithms and then, we compare the results with ProSOM method [5]. We use two different datasets which are small and large datasets. We show that f-measure of ProSOM decreases when the number of sequences in dataset increases. Furthermore, ProSVM gives better result than ProSOM algorithm on large dataset. Only structural features are used in these algorithms.

Secondly, we introduced a new hierarchical prediction system called 3S1C (Signal, Structure, Similarity, and Context Features). We introduce similarity features and we show that similarity features increase overall accuracy. We use support vector machine to classify signal, context, structure, and similarity features sets separately. Finally, we combine the probabilities coming from classification of these features by using Multilayer Perceptron. Looking at the current promoter prediction methods, MetaProm uses neural network for human promoter prediction. MetaProm integrates the prediction of current promoter prediction programs. However, MetaProm aims to find the promoter regions associated only CpG islands. On the other hand, 3S1C is not limited to such regions.

1.3 Organization of the Thesis

The rest of the thesis is organized as follows:

- In Chapter 2, we give the details of existing methods for promoter prediction.
- In Chapter 3, proposed methods in this study, which are ProK-means, ProSVM, and 3S1C methods are introduced.
- In Chapter 4, results of ProK-means, ProSVM, and 3S1C algorithms are given and evaluated.
- In Chapter 5, we conclude with a summary and future directions.

- In Appendix A, the detailed information about support vector machines and multilayer perceptron is given.
- In Appendix B, complete results related to ProK-means are given.
- In Appendix C, glossary of the terms used in the thesis is listed.

CHAPTER 2

RELATED WORK

2.1 Promoter Classification in the Literature

In the literature, there are a lot of methods about classification or recognition of promoter regions in human sequences. This section provides a review of the following topics:

- promoter sequence databases
- features used in promoter classification problems
- classification methods used in promoter classification

There are a lot of databases including promoter sequences in the literature. In this part, information about three databases currently used and publicly available is given. They are the Eukaryotic Promoter Database (EPD), the Ensemble database, and Database of Transcription Start Site (DBTSS).

2.1.1 Promoter Sequence Databases

The Eukaryotic Promoter Database (EPD) [6] is a database of promoters which determine a gene region nearly upstream of a transcription initiation site. EPD was planned for comparing sequences and played an important role for development of the eukaryotic promoter prediction algorithms. EPD is based on experimental data which determines transcription start sites of eukaryotic genes. This is the main goal of EPD. EPD links to other databases. By means of machine readable promoters which are provided from EMBL [7] nucleotide

sequence database, the promoter sequences are accessed. EPD is a strictly non-redundant database and one sequence corresponds to one transcription start site in a genome.

The Ensemble database [8] supplies extensive source of automatic annotation of large genome sequences. Therefore, it supplies a framework to organize the sequences of large genomes [9]. Ensemble provides also flexible and reliable variety of genomes. Additionally, it is a high performance and highly scalable database. The Ensemble database plays an important role on gene annotation and comparison analysis. In comparison analysis, protein similarity, pairwise similarity alignment and transcript similarity of orthologous are included [8] . Briefly, Ensemble provides high coverage but mixed quality regions around gene starts [10]. For detailed information about the Ensemble database, look at http://www.ensembl.org.

DataBase of Transcription Start Sites (DBTSS) [11] is a collection and analysis of eukaryotic promoter regions. It contains both the information of transcription start site and the sequence similarity between the upstream regions of orthologous genes [12]. Eukaryotic Promoter Database (EPD) provides reliable promoter sequences but the number of these sequences is not enough for promoter analysis [12]. To solve this problem, DBTSS were developed in 2002. Advantage of DBTSS is that the distribution of transcription start site is shown for each gene. Briefly, DBTSS gives the reliable promoter sequences which have best coverage. Therefore, most of the current promoter prediction methods use the sequences provided from the DBTSS database.

2.1.2 Features used in promoter classification problems

One of the important problems in promoter classification is to extract informative features to separate the sequences of promoters from non promoters. Zeng *et. al.* [3] investigated 66 papers published between 1998 and 2008 about human promoter classification. In these studies, signal, context, and structure features are used to determine core promoter regions.

Signal features are powerful signals which reflect the local properties of promoter sequences. Figure 2.1 gives the distribution of articles related to signal features published per year. When looking at the figure, using of signal features in promoter classification decreases after 2006. Signal features were commonly studied between 2000 to 2006. In such studies, for detection of core promoters, the TATA box, initiator, TFIIB recognition element were used [3, 13].

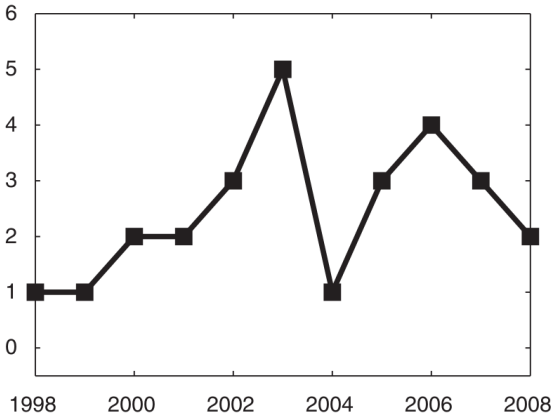

Figure 2.1: The distribution of articles related to signal features published per year (Adapted from [3])

Since 2007, CpG island features have been investigated to extract signal features. The reason is that human promoters have a high CpG content because 3′ end promoters are less probably to be located within CpG islands than 5′ end promoters [3, 14]. The "p" in CpG refers to phosphodiester bound between the cytosine and the guanine [1].

CpG island is a region of DNA sequence in about 200 bp length which is G+C content enrich. There are two important CpG features used for promoter recognition:

$$GC_p = p(C) + p(G) \tag{2.1}$$

$$\frac{C_p G_o}{e} = \frac{p(CG)}{p(C)p(G)} \tag{2.2}$$

where p(CG), p(C) and p(G) are the percentages of CG, C, and G, respectively in a nucleotide sequence [10]. CpG signals are very effective if CpG poor and CpG rich promoters are together.

Context features provide information about content of the genome and are generally represented by n-base-long nucleotide sequences [3]. A set of n-mer (n length nucleotide sequence) is predicted from a training set in order to extract context features. Additionally, the distribution of the n-mer may introduce new promoters whose details are not known. That is why context feature can help reduce false positive rates. Figure 2.2 shows the distribution of articles related to context features published per year. Using of the context features in promoter classification problems increased drastically between 2004 and 2006 and continue to increase recently.

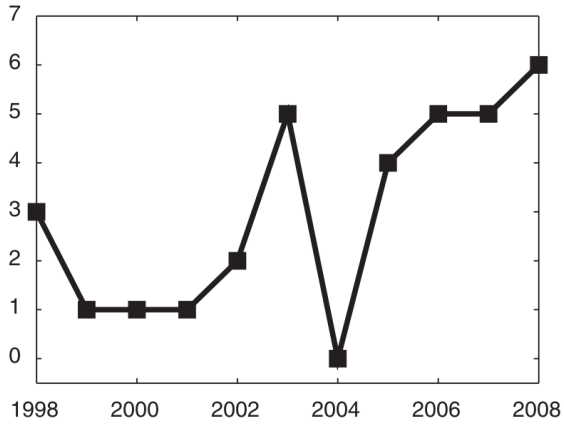

Figure 2.2: The distribution of articles related to context features published per year (Adapted from [3])

Structural features are based on physical properties of the DNA three dimensional structures. Figure 2.3 shows the distribution of articles related to structure features published per year.

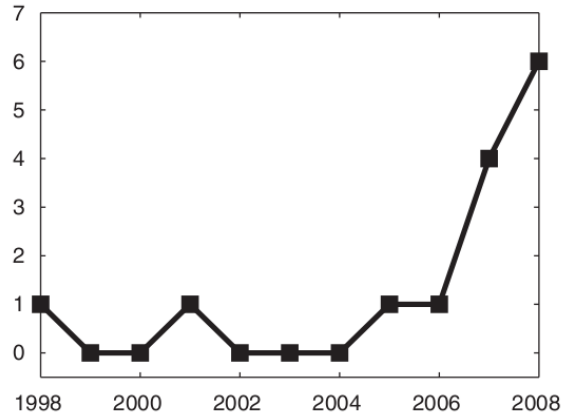

Figure 2.3: The distribution of articles related to structure features published per year (Adapted from [3])

Using structure features in promoter classification strongly increased between 2006 and 2008. Structure features have been popular since 2007. Three types of structure features in human core promoter regions were effectively used in these studies. The first is tetranucleotide potential energy surface model [10, 15]. DNA flexible parameters derived from long atomic dynamics simulations in water is the second [16]. Base stacking property of DNA is the last [5, 10]. In these studies, in order to calculate structure features, base stacking energy from the properties of DNA three dimensional structure is used. Sequences are converted into numerical values via base stacking energy tables. High values in the tables show regions

which melt easily [5, 17]. There are dinucleotide, trinucleotide, and tetranucleotide conversion tables used to extract structural features. Tetranucleotides supply more genomic content information than dinucleotide and trinucleotide features [3].

2.1.3 Classification methods used in promoter classification

Using appropriate classifiers to separate promoter classes from non promoter classes is an important problem as the feature extraction problem in promoter classification. Looking at the literature, current methods propose to use integrated methods in promoter classification. Three major classification methods which have been used currently in promoter classification problems are listed below [3]:

- The discriminative model which uses an optimal threshold or classification boundary in signal, context, and structure feature space. For instance, artificial neural networks (ANN), discriminative functions, and support vector machines (SVMs)
- The generative model that defines the generative process of signal, context, and structure observations. For instance, Position weight matrix (PWM), nearest neighborhood, and hidden Markov models (HMM)
- The integration of classifier techniques

In promoter classification problems, combined methods have become popular lately. The information about these methods are given in the following.

PromoterExplorer [18] is a method which uses CpG signals and distribution of 5-mer features. A high dimensional vector is obtained by combining them to improve the performance. Therefore, it combines signal and context features. To select most informative and discriminative features, the AdaBoost algorithm is used.

CoreBoost [19] develops a boosting technique with stumps for selecting the features from CpG signals, Inr and TATA box. It uses mechanical properties and sequence features from Markovian modeling. Additionally, frequency of n-mers is used as feature.

SCS [10] integrates signal, context, and structure features by using decision trees. They investigate CpG islands to extract signal features and signal features are classified using Gaussian

Mixture Models. Second, they looked at n-mer frequency for extracting context feature and they used Naive Bayes Classifier for classifying context features. Finally, to extract structure features, base stacking energy tables were used and local regions are selected to generate structure features. They used Gaussian Mixture Models for classification of the structure features. 251 bps sequences are used and these sequences are provided by the DBTSS database.

MetaProm [14] uses Multilayer Perceptron function in the Weka [20]. It integrates predictions coming from PSPA, McPromoter, DragonGSF, FirstEF, FProm, and DragonPF classifiers. It uses artificial neural network, and it integrates signal, context, and structure features with an indirect way. This combination improved overall accuracy. The DBTSS database is used for promoter sequences.

Similarly, EnsemPro [21] integrates classification results coming from Eponine, NNPP, FirstEF, Promoter 2.0, and DPE classifiers. They used three methods which are majority voting, the weighted voting, and Bayesian approach to integrate the classifiers.

2.2 Differences of this study from existing methods

ProK-means and ProSVM algorithms are compared with ProSOM algorithm. Differences are listed below:

- ProSOM uses only dinucleotide conversion. However, both dinucleotide and trinucleotide conversions are applied to ProK-means and ProSVM algorithms.
- ProSOM uses 3 bps sliding window, on the other hand, ProK-means and ProSVM algorithms use 5 bps sliding window.
- ProSOM uses 251 bps sequences, on the other hand, ProK-means and ProSVM are implemented on datasets which include 251 bps and 400 bps sequences separately.

Then, we show that the ProSVM algorithm provides better results than the ProSOM algorithms.

Existing methods use signal, context, and structure features and combine them into a system. In addition to these, similarity features are used in this study. In the literature, when we look

at promoter sequence similarity between human and other species, human-mouse genome comparison was done in 2000 by Wasserman et. al. [22]. They used the Bayes block assigner (BBA) which focuses on aligning highly conserved, ungapped reserved parts in regulatory elements. Their individual alignments are based on conserved blocks. Moreover, they used PAM matrix for nucleotide comparison.

In this study, it is checked whether promoter regions of human is similar to promoter regions of C. merolae, mouse, zebrafish, and malaria species. This comparison is performed by using similarity feature defined in Chapter 3. Human sequences with non human promoter sequences are compared using the similarity feature. As a results of this comparison, a similarity score is assigned to human promoter and non promoter sequences. These scores and the classification results show that promoter regions of human sequences are similar to promoter regions of non human promoter species' (C. Merolae, mouse, zebrafish, and malaria).

Furthermore, signal, context, structure, and similarity feature sets are classified by using Support Vector Machines (SVM) and the probabilities coming from classification results of these feature sets are combined by using Multilayer Perceptron (MLP). When looking at the literature, MetaProm integrates results of various promoter prediction programs to find promoters associated with CpG islands by using neural network. That is, MetaProm focuses on the promoters associated with CpG islands. On the other hand, 3S1C is not limited the promoters located in CpG island.

CHAPTER 3

METHODS

In order to separate promoter sequences from non promoter sequences, ProK-means, ProSVM, and 3S1C algorithms are proposed in this section. Furthermore, using the conservation of the genomic data, similarity features are introduced.

In ProK-means and ProSVM algorithm, structural features of DNA sequences are used, on the other hand, 3S1C uses combination of signal, context, structure, and similarity features of DNA sequences.

In this section, firstly, ProK-means, ProSVM methods and datasets of these methods are introduced and then, 3S1C method and its dataset are introduced.

3.1 PROK-MEANS AND PROSVM ALGORITHMS

ProK-means and ProSVM algorithms, which are unsupervised and supervised methods respectively, are introduced to distinguish core promoter regions. ProK-means and ProSVM are constructed by inspiring ProSOM algorithm, which is an existing promoter prediction method. Therefore, information about ProSOM [5] and some details are given before introducing ProK-means and ProSVM algorithm in order to understand these methods easily.

3.1.1 The ProSOM Algorithm

ProSOM [5], which uses an unsupervised self-organizing map (SOM), is a promoter prediction technique. It uses promoter sequences as positive data, and intergenic and transcribed sequences as negative data. The promoter sequences are extracted from the region [-200,

50] around the TSS. To extract features, the structural profile of a set of DNA sequences is calculated with the following steps:

- The nucleotide sequence is converted into a sequence of numbers by replacing each dinucleotide with its energy value.
- For each position, the average of the sequences is taken for that position.
- 3 bps sliding window is used for each sequence to avoid noise.

3.1.2 Datasets

In this part, datasets used in ProK-means and ProSVM algorithm are introduced. Two different datasets are used in order to observe the performance changes of the methods. These datasets are called as Dataset 1 and Dataset 2. Promoter sequences are used as positive data, but transcribed and intergenic sequences are used as negative data. Promoter sequences are extracted from the DBTSS database [23]. H. sapiens is chosen as sequence type. Details of Dataset 1 and Dataset 2 are summarized in the following:

- In Dataset 1, downstream and upstream are set to 200 in the DBTSS database, so all sequences are extracted from the region [-200,200] around the TSS. That is, the sequences are 401 bps in length. 800 promoter, 800 intergenic, and 800 transcript sequences are used totally. 700 promoter, 700 intergenic, and 700 transcript sequences are used for training and the others are used for testing.
- In Dataset 2, the sequences located the region [-200,50] around the TSS are provided by ProSOM [5]. That is, the sequences are 251 bps in length. 25000 promoter, 25000 intergenic, and 25000 transcript sequences are used totally. 15000 promoter, 15000 intergenic, and 15000 transcript sequences are used for training and the others are used for testing.

3.1.3 Structural Profile

In this section, features which are used by ProSVM and ProK-means algorithms are introduced. To extract the features, structural profiles are calculated. Two different implementa-

tions of structural profiles are used in ProK-means and ProSVM methods.

In the first implementation, the sequences are converted into numerical values which represent their energy values by using dinucleotide conversion table [24] shown in Table 3.1. Dinucleotide conversion is abbreviated as DiCon in this study.

Table 3.1: Dinucleotide Conversion Table (Adapted from [24])

Dinucleotide	Base Stacking Energy	Dinucleotide	Base Stacking Energy
AA	-5.37	GA	-9.81
AC	-10.51	GC	-14.59
AG	-6.78	GG	-8.26
AT	-6.57	GT	-10.51
CA	-6.57	TA	-3.82
CC	-8.26	TC	-9.81
CG	-9.69	TG	-6.57
CT	-6.78	TT	-5.37

After that, the sequences are smoothed by using 5 bps sliding window for noise reduction. 5 bps sliding window is illustrated in Figure 3.1.

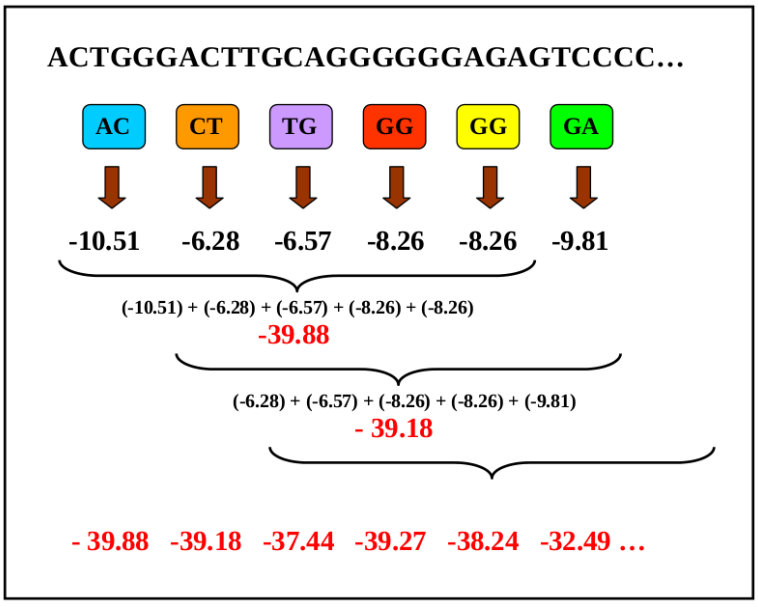

Figure 3.1: 5 bps sliding window

In the second implementation, trinucleotide conversion [25], which is shown in Table 3.2, in-

stead of dinucleotide conversion is used and then 5 bps sliding window is used. Trinucleotide conversion is abbreviated as TriCon in this study.

Table 3.2: Trinucleotide Conversion Table (Adapted from [25, 26]). (The abbreviations are TriCon: Trinucleotide conversion, BSE: Base Stacking Energy)

TriCon	BSE	TriCon	BSE	TriCon	BSE	TriCon	BSE
AAT	-0.280	CGC	-0.077	CGA	-0.003	TAA	0.068
ATT	-0.280	GCG	-0.077	TCG	-0.003	TTA	0.068
AAA	-0.274	AGG	-0.057	GGA	0.013	GCA	0.076
TTT	-0.274	CCT	-0.057	TCC	0.013	TGC	0.076
CCA	-0.246	GAA	-0.037	CAA	0.015	CTA	0.090
TGG	-0.246	TTC	-0.037	TTG	0.015	TAG	0.090
AAC	-0.205	ACG	-0.033	AGC	0.017	GCC	0.107
GTT	-0.205	CGT	-0.033	GCT	0.017	GGC	0.107
ACT	-0.183	ACC	-0.032	GTA	0.025	ATG	0.134
AGT	-0.183	GGT	-0.032	TAC	0.025	CAT	0.134
CCG	-0.136	GAC	-0.013	AGA	0.027	CAG	0.175
CGG	-0.136	GTC	-0.013	TCT	0.027	CTG	0.175
ATC	-0.110	CCC	-0.012	CTC	0.031	ATA	0.182
GAT	-0.110	GGG	-0.012	GAG	0.031	TAT	0.182
AAG	-0.081	ACA	-0.006	CAC	0.040	TCA	0.194
CTT	-0.081	TGT	-0.006	GTG	0.040	TGA	0.194

3.1.4 ProK-means Algorithm

After obtaining feature sets according to mentioned above rules, k-means clustering algorithm is run for various k values. Therefore, we obtain k clusters which have a combination of promoter and non promoter sequences. In order to determine promoter clusters, we used a user defined threshold. The threshold is determined by using cluster probabilities which are the ratio of number of promoter sequences to total number of sequences in each class. Therefore, a cluster probability is assigned to a cluster and we obtain k cluster probabilities. Looking at these probabilities, the threshold is determined. In order to obtain threshold, firstly, we choose the highest cluster probability as a threshold and we measure the performance of the system and then we choose second highest cluster probability as a threshold and we measure the performance of the system, and we repeat this process until the system reaches the highest performance. If the probability of a cluster is larger than the threshold, this cluster is called as a promoter region (or cluster). Otherwise, it is called as a non promoter region.

3.1.5 ProSVM Algorithm

After constructing feature sets, support vector machine is used to separate promoters from non promoters. Support vector machines separate the data by using kernel functions. In this study, linear, polynomial, radial basis function (RBF), and sigmoid kernels are used. The detailed information about support vector machines is given in Appendix A.

3.2 3S1C ALGORITHM

A two-level classification system called 3S1C is introduced to classify promoter regions. In the first level of the system, signal, context, structure and similarity features are extracted from DNA sequences and classified by using Support Vector Machines (SVM). Then, in the second level of the system, these features are combined with multilayer perceptron in a hierarchical manner. Figure 3.2 illustrates the architecture of the 3S1C algorithm.

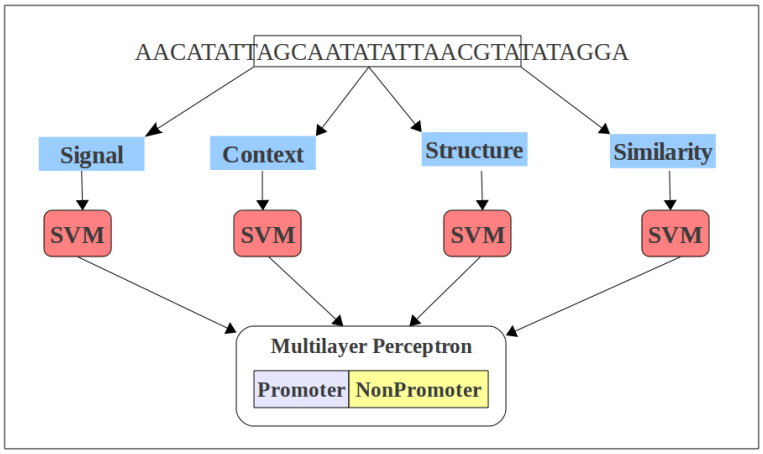

Figure 3.2: The architecture of 3S1C algorithm

3.2.1 Dataset

In this section, promoter and non promoter sequences are used for prediction of human genome promoters. Promoter sequences consist of human, C. merolae, zebrafish, mouse, malaria species promoters. We used the non human promoter sequences for only computing the similarity features. However, non promoter sequences consist of intron, exon, and 3′ UTR

sequences.

Human promoter and non-promoter sequences are provided by J. Zeng [10]. C. merolae, zebrafish, mouse, malaria promoter sequences are obtained from the DBTSS database [23]. The reason for choosing these types of organisms is that the DBTSS database includes them. These four species are refered as *other species* in the following parts.

All sequences are 251 base pairs (bps) in length. Promoter sequences are extracted between -200 to 50 around Transcription Start Site (TSS), and non promoter sequences are obtained from random locations from intronic, exonic, and 3′ UTR regions with 251 bps in length. In the first level classification, dataset is divided into two equal parts, training and testing and given in Table 3.3. Then, in second level classification, the data is divided into five parts because five-fold cross validation is used to evaluate the system.

Table 3.3: Number of promoter and non-promoter sequences used in Datasets

Datasets	Number of promoter sequences	Number of non-promoter sequences
Train Data	15473	46419
Test Data	15473	46419

3.2.2 Data Preparation for the First Level Classification

Signal, context, structure, and similarity features are classified separately in the first level of the algorithm. In order to extract signal, context, structure, and similarity features, promoter sequences are used as positive data and non promoter sequences as negative data. The number of sequences used in training and testing is given Table 3.4. To calculate similarity features, we looked at the similarity between human sequences and non human promoter sequences. Species and the number of sequences used to extract the features are given in Table 3.5.

3.2.3 Data Preparation for the Second Level Classification

In the second level of 3S1C algorithm, classification results of four classifiers coming from the first level of the algorithm are used.

In the first level, features are classified by Support Vector Machine (SVM) separately. Two

Table 3.4: The number of sequences used in the first level classification

	Training		Testing		
Features	Promoter	Non Promoter	Promoter	Non Promoter	Total
Signal	15473	46419	15473	46419	123784
Context	15473	46419	15473	46419	123784
Structure	15473	46419	15473	46419	123784
Similarity	30946	46419	30946	46419	154730

Table 3.5: Species and the number of promoter and non promoter sequences in the first level

Promoter					Non Promoter		
Human	C. Merolae	Mouse	Zebrafish	Malaria	Exon	Intron	3′ UTR
15473	2500	5207	6373	1393	15473	15473	15473

probabilities, which are promoter and non promoter probability, for each sequence are obtained as the result of first level classification. Promoter probability is used as the input data for the second level.

After classification of each test data in the first level classifiers, four labels are obtained for a sequence coming from signal, context, structure, and similarity features and these labels are assigned as "1" or "2" by SVM. Label "1" stands for human promoter and Label "2" stands for human non-promoter sequence. Multilayer Perceptron is used in order to assign the final label.

In this level, 15473 promoter sequences and 46419 non promoter sequences are used. The dataset is separated into five parts and five-fold cross validation is applied to evaluate performance of the second level classifier.

3.2.4 Feature Extraction

Feature extraction is a critical and important step for determining core-promoter regions and it affects the performance of the system. Features should be chosen as to well-characterize the data because the characteristic of the promoters show some differences e.g. some promoters are located around TSS, on the other hand, some promoters are located hundreds bps before TSS.

After constructing the dataset as described in the dataset part, signal, context, structure, and similarity features are extracted. Although these features are extracted from the same sequences and they affect each other, we treat them as they are statistically independent.

3.2.4.1 Signal Features

Signal features are powerful signals which reflect the local properties of promoter sequences. CpG island features have been investigated to extract signal features by using Equations 3.1 and 3.2. The reason is that 3′ end promoters are less probable to be located within CpG islands than 5′ end promoters [3, 14].

$$GC_p = p(C) + p(G) \tag{3.1}$$

$$\frac{C_pG_o}{e} = \frac{p(CG)}{p(C)p(G)} \tag{3.2}$$

where p(CG), p(C) and p(G) are percentage of CG, C and G, respectively in a nucleotide sequence [10].

These values are calculated for each sequence in the dataset, and they constitute two dimensions in the feature space.

3.2.4.2 Context Features

Context features provide information about content of the genome and are represented by frequencies of n-base-long nucleotides. Context features may include signal features. For example, TATA box, Inr such as CCAT, and CpG islands such as CGGC is 4-mer. Additionally, the distribution of the n-mer may introduce new promoters whose details are not known. That is why context features can help reduce false positive rates.

Context features are extracted as frequencies of dinucleotides and 16-dimensional vectors, $(c_1, c_2, ..., c_{16})$, are obtained for each sequence. Frequencies of occurrence of dinucleotides are calculated by using equation 3.3.

$$c_i = \frac{f_i}{s_i - 1} \tag{3.3}$$

where c_i is i^{th} context feature, f_i is the frequency of i^{th} sequence, and s_i is the length of i^{th} sequence.

3.2.4.3 Structure Features

Structure features are based on physical properties of the DNA three dimensional structures. One of these properties is base stacking energy.

To extract structure features, sequences are converted into numerical values via base stacking energy table shown in Table 3.1. There are 16 values for dinucleotide conversion and high values in the tables show regions which melt easily [5, 17]. There may be some missing nucleotides in the sequences. To overcome this problem, sequences are smoothed by using 5 bps sliding window as illustrated in Figure 3.1. As a result of this conversion, 251 numeric values are obtained because the length of sequences in dataset is 251 bps. The profile seems much noisy without any regularity, so six specific positions are determined to construct structure features. These values are shown as $(s_1, s_2, ..., s_{16})$ where s_1 is the average value from location -200 to location -151, s_2 is the value at location -32, s_3 is the value at location -28, s_4 is the value at location -3, s_5 is the value at location 0, and s_6 is the average value from location +11 to location +30 [10]. These feature vectors reflect the local changes around TSS.

3.2.4.4 Similarity Features

Although evolution has led to the specification of all living organisms, most of the living organisms' genomes show high amount of similarities because of conservation of the genetic data from generation to generation. Similarity feature is based on the conservation of the genetic data. Promoter regions of human beings might be similar to promoter regions of other species. It is better to check whether sequence of human in question resembles to any of the experimentally verified promoter region of other species.

To extract similarity features, human promoter and non promoter sequences are compared with C. Merolae, mouse, zebrafish and malaria species' promoters. A similarity score is assigned to each promoter and non promoter sequences. This score determines how much similar promoters of human and other types are. Similarity features are defined by this score. In

order to assign a similarity score to a human sequence, the human sequence is compared with all promoter sequences of other species. The highest score of comparison result is assigned as similarity score.

In order to find similarity score between two sequences, the total number of nucleotides which are identical and located in the same position is calculated.

3.2.5 3S1C Algorithm

3S1C (Signal, Structure, Similarity, and Context) is a 2-level hierarchical classification model which combine Support Vector Machines and Multilayer Perceptron classification methods.

In the first level, after constructing signal, context, structure, and similarity feature sets in the previous section, they are classified independently and combined in this part. Support Vector Machines are chosen for classification of feature sets. The reasons for choosing Support Vector Machines are:

- SVM can separate the feature spaces in two parts even there is non-linear relationship between the variables.
- SVM is a sparse kernel method, that is, it does not use whole data, it holds certain part of the data in the memory. Data is only represented as support vectors.
- SVM works in infinite dimensional Hilbert Spaces and benefits from kernels in Hilbert Space.

LIBSVM 3.1 [27], developed by Chang and Lin, is chosen for the first level classification. We obtain four component classifiers by classifying signal, context, structure, and similarity features separately. For each component classifier, we apply the rules listed below.

- SVM is trained with respect to four different kernels which are linear, polynomial, radial basis function (RBF) and sigmoid.
- In the training part of first level classifiers, -b parameter is used to obtain a probability estimate of belonging to a class for each sequence.
- Accuracy (Equation 3.4) is calculated for each kernel type.

- The classifier which has the highest accuracy is used for the final classification decision.

$$Accuracy = \frac{Number\ of\ sequences\ correctly\ predicted\ in\ test\ data}{Number\ of\ total\ sequences\ in\ test\ data} \tag{3.4}$$

In the second level of classification, classifiers are combined in order to improve the overall accuracy of the classification. The output probabilities coming from the component classifiers are used for the final classification decision. There are two probabilities coming from a component classifier for a sequence. The probability belonging to the promoter class is used as features in second level classifiers. Then, feature set generated by combining probabilities coming from component classifiers is classified by using Multilayer Perceptron. Therefore, a 2-level hierarchical system is constructed. The reasons for selecting Multilayer Perceptron are :

- When we look at the literature, multilayer perceptron is proposed as a special model which obtains the satisfactory thresholds or classification boundaries in the signal, context, and structure feature space.
- Multilayer Perceptron can model data which has nonlinear relationship between variables fast.
- Multilayer Perceptron can handle interactions between variables.

CHAPTER 4

RESULTS

In this part, the results of the ProK-means, ProSVM, and 3S1C methods are given and evaluated. ProK-means and ProSVM methods are applied on Dataset 1 and Dataset 2 which are explained in Chapter 3. The results of ProK-means and ProSVM algorithms are compared with an existing method ProSOM. Then, the results of 3S1C algorithm are given. Some terms which are used to evaluate the algorithms are given in the following.

Confusion matrix shown in Figure 4.1 gives information about actual and predicted classification done by the classifier. In the table, True Positives (TP) is the number of correctly classified promoters, True Negative (TN) is the number of correctly classified non promoters, False Positive (FP) is the number of misclassified promoters, and False Negative (FN) is the number of misclassified non promoters.

		Actual Outcome		
		True	False	
Test outcome	Positive	True Positive	False Positive	Positive predictive value
	Negative	False Negative	True Negative	Negative predictive value
		↓ Sensitivity	↓ Specificity	Accuracy

Figure 4.1: Confusion Matrix (Adapted from [1])

Precision, recall and F-measure (harmonic mean of precision and recall) are used in order to evaluate performance of ProK-means and ProSVM algorithm. The formulas related to these measurements are given in the following.

Precision is calculated by using Equation 4.1. It gives the ratio of correctly classified promoters to sum of correctly and incorrectly classified promoters in the test dataset.

$$precision = \frac{TP}{TP + FP} \tag{4.1}$$

Recall (sometimes called sensitivity) is calculated by using Equation 4.2. It gives the ratio of correctly classified promoters to sum of correctly classified promoters and incorrectly classifies non promoters in the test dataset.

$$recall = \frac{TP}{TP + FN} \tag{4.2}$$

Specificity is calculated by using Equation 4.3. It gives the ratio of correctly classified non promoter to sum of correctly classified non promoters and incorrectly classified promoters in the test dataset.

$$specificity = \frac{TN}{TN + FP} \tag{4.3}$$

Accuracy is calculated by using Equation 4.4. It gives the ratio of sum of correctly classified promoter and non promoter sequences to all sequences in the test dataset.

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{4.4}$$

4.1 ProK-MEANS RESULTS

ProK-means results are evaluated according to dinucleotide and trinucleotide conversions separately. ProK-means method is implemented by using MATLAB and it suffers from the size constraints on the number of sequences. Because of this drawback, we used small amount of sequences. We can apply ProK-means algorithm only Dataset 1. We used Microsoft Excel and C++ on Linux platform for preparation of the structural profiles. ProK-means results on Dataset 1 are shown in the following.

4.1.1 Results for Dinucleotide Conversion

Number of clusters (i.e. k values) are set as 16, 25, 36, and 49 in the experiments. The reason for this, we obtained the best f-measure values with respect to these numbers. We used the trained k-means to predict promoter regions. We applied the same method described in ProSOM [5]. We attached a probability to each cluster. If the structural profile of a sequence maps to a cluster that has a probability equal to or above a user defined threshold, we predict it as a promoter region.

We tested the algorithm according to various thresholds. Threshold values are set by looking at the cluster probabilities. First, we took the highest cluster probability as the first threshold. If the probability of a region is higher than the threshold, this region was decided as the promoter region, otherwise, non promoter region. Then precision, recall and f-measure are calculated for this threshold. After these calculations, we choose second highest probability as a threshold. Then, we perform the same calculations until obtaining the highest f-measure value. This validation method is applied for the conditions k=16, 25, 36, and 49. The highest f-measure values for k=16, 25, 36 and 49 are 0.823, 0.831, 0.832, and 0.817 respectively. For k = 36, we obtained the highest F-measure value among all clusters. Table 4.1 shows choosing of the threshold.

Table 4.1: Dinucleotide conversion validation results for k=36

Threshold	Precision	Recall	F-measure
1.000	1.000	0.019	0.038
0.950	1.000	0.240	0.388
0.930	1.000	0.337	0.504
0.880	1.000	0.394	0.566
0.860	1.000	0.413	0.585
0.840	0.982	0.538	0.696
0.770	0.969	0.606	0.746
0.750	0.956	0.625	0.756
0.480	0.854	0.731	0.788
0.340	0.809	0.856	**0.832**

In order to observe the change of f-measure according to the threshold, see Figure 4.2. Therefore, we determine the promoter regions as choosing k=36 for dinucleotide conversion (DiNuc). Detailed results related to ProK-means for dinucleotide conversion algorithm are shown in Appendix B.

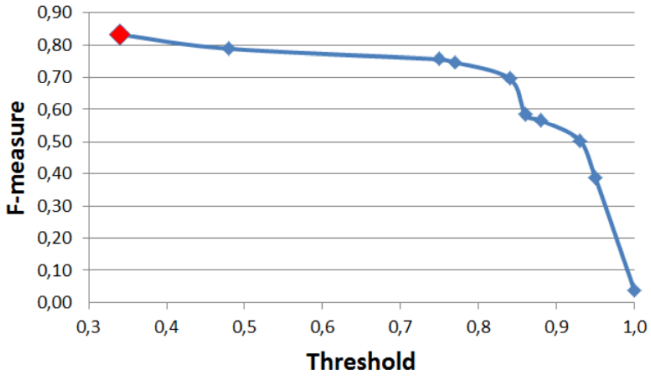

Figure 4.2: DiCon: Relationship between threshold and f-measure for k=36

4.1.2 Results for Trinucleotide Conversion

We used the same method with dinucleotide conversion. However, there is small difference between them. Trinucleotide feature vectors are used instead of dinucleotide feature vectors. The method is applied for k=16, 25, 36, and 49. The highest f-measure values for k=16, 25, 36, and 49 are 0.664, 0.667, 0.637, and 0.602 respectively. We obtained the highest f-measure value among all clusters for k = 25. Table 4.2 shows choosing of the threshold and the relationship between threshold and f-measure are shown in Figure 4.3. Therefore, we determine the promoter regions as choosing k=25 for trinucleotide conversion. Detailed results related to ProK-means for trinucleotide conversion algorithm are shown in Appendix B.

Table 4.2: Trinucleotide conversion validation results for k=25

Threshold	Precision	Recall	F-measure
0.635	0.543	0.240	0.333
0.498	0.602	0.538	0.569
0.473	0.613	0.731	**0.667**
0.381	0.566	0.788	0.659
0.349	0.547	0.837	0.662
0.286	0.520	0.856	0.647

When looking at the F-measure values for both dinucleotide and trinucleotide conversions, dinucleotide conversion has higher F-measure values than trinucleotide conversion.

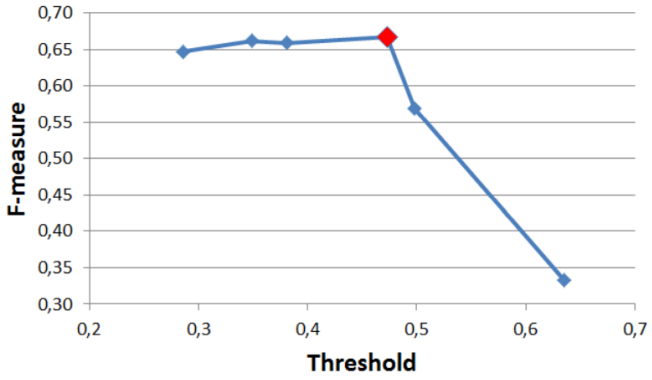

Figure 4.3: TriCon: Relationship between threshold and f-measure for k=25

4.2 ProSVM RESULTS

We apply ProSVM methods on Dataset 1 and Dataset 2 for dinucleotide and trinucletide conversion. The implementation is done by using SVM^{light} [28] program.

4.2.1 ProSVM Results on Dataset 1

We train an SVM model with 700 promoters from the DBTSS database, 700 intergenic, and 700 transcript sequences by using SVM^{light} [28] program on Dataset 1. 100 promoters, 100 intergenic, and 100 transcript sequences are used for testing. The program has kernel options. We classified the data using linear, polynomial, RBF, and sigmoid kernels with default parameters. Table 4.3 and Table 4.4 shows the testing results for dinucleotide and trinucleotide conversions respectively. The results of sigmoid kernel for dinucleotide conversion are not shown in Table 4.3 because f-measure value for this kernel is very low, that is , it is nearly zero.

Table 4.3: Dinucleotide conversion validation results for SVM on Dataset 1

Kernel	Precision	Recall	F-Score
Linear	0.373	0.562	0.449
Polynomial	0.861	1.000	0.925
RBF	1.000	1.000	**1.000**

Both dinucleotide and trinucleotide conversions give best result for RBF kernel on Dataset 1.

Table 4.4: Trinucleotide conversion validation results for SVM on Dataset 1

Kernel	Precision	Recall	F-Score
Linear	0.431	0.821	0.565
Polynomial	0.574	1.000	0.729
RBF	1.000	1.000	**1.000**
Sigmoid	0.343	0.991	0.510

4.2.2 ProSVM Results on Dataset 2

ProSVM algorithm is applied on Dataset 2 for both dinucleotide and trinucleotide conversion. Table 4.5 shows ProSVM results for dinucleotide conversion on Dataset 2. We run ProSVM algorithm by choosing linear, polynomial, RBF, and sigmoid kernels. F-measure values for RBF and sigmoid kernels are nearly zero, that is, they have very low f-measure, so results of sigmoid and RBF kernels are not shown in Table 4.5.

Table 4.5: ProSVM validation results for dinucleotide conversion on Dataset 2

Kernel	Precision	Recall	F-Score
Linear	0.199	0.360	0.256
Polynomial	0.890	0.307	**0.457**

Table 4.6 shows ProSVM results for trinucleotide conversion on Dataset 2. We obtained the best results using the polynomial kernel for dinucleotide and trinucleotide conversion. We obtained better results for trinucleotide conversion than dinucleotide conversion.

Table 4.6: ProSVM validation results for trinucleotide conversion on Dataset 2

Kernel	Precision	Recall	F-Score
Linear	0.282	0.120	0.168
Polynomial	0.463	0.534	**0.496**
RBF	1.000	0.000	0.000
Sigmoid	0.324	0.250	0.282

4.3 ProSOM RESULTS

We run the ProSOM algorithm on Dataset 1 and Dataset 2. Table 4.7 shows the ProSOM results. ProSOM gives better results on Dataset 1. If the number of elements in the dataset gets higher, ProSOM results in lower f-measure.

Table 4.7: ProSOM Results

Datasets	Precision	Recall	F-Score
Dataset 1	0.946	0.389	0.554
Dataset 2	0.698	0.204	0.315

4.3.1 Comparison of the Methods on Dataset 1

Table 4.8 gives detailed comparison of the three methods on Dataset 1. ProSVM with RBF kernel gives better results than ProK-means and ProSOM methods on Dataset 1.

Table 4.8: Comparison of three methods on Dataset 1

Methods	Precision	Recall	F-Score
ProK-means	0.809	0.856	0.832
ProSVM	1.000	1.000	**1.000**
ProSOM	0.946	0.389	0.554

4.3.2 Comparison of the Methods on Dataset 2

Table 4.9 shows the comparison results of the two methods on Dataset 2. ProSVM gives better results with polynomial kernel for trinucleotide conversion than the ProSOM algorithm.

Table 4.9: Comparison of two methods on Dataset 2

Methods	Precision	Recall	F-Score
ProSOM	0.698	0.204	0.315
ProSVM	0.463	0.534	**0.496**

In conlusion, ProSVM with RBF kernel for dinucleotide and trinucleotide conversion gives better results than ProK-means and ProSOM methods on Dataset 1 (small dataset). Addition-

ally, ProSVM with polynomial kernel for trinucleotide conversion on Dataset 2 (large dataset) gives better results than ProSOM.

4.4 3S1C RESULTS

To measure the performance of the system, five fold cross validation is applied. In this validation, data is separated into five groups for training and testing. One fold is used for testing and remaining four folds are used for training. This process is iterated until each group is tested once. To measure the performance of the system, sensitivity, specificity, and accuracy are calculated.

LIBSVM 3.1 [27] is used for first level classification and Weka 3.6.4 [20] is used for second level of classification. We used Microsoft Excel and C++ on Linux platform for preparation of the datasets for the first and second levels of classification.

4.4.1 The Results of the First and Second Level Classifiers

In order to evaluate first level classifiers, each component classifier is tested by using linear, polynomial, radial basis, and sigmoid kernels. Accuracy (Equation 4.4) is calculated for each classifier in order to find the best kernel classifier among each component classifier.

Table 4.10: Accuracy results of the first level classsifiers

Features	Linear	Polynomial	RBF	Sigmoid
Signal	0.7906	0.7864	0.7879	0.7885
Context	0.7932	0.7855	0.7928	0.7925
Structure	-	-	0.7747	0.7500
Similarity	0.7500	-	0.7519	0.7500

Accuracy results are shown in Table 4.10. Missing values are shown as " - ". The reason is that SVM does not converge, in other words, SVM does not find an optimal separating hyperplane with such kernels. We obtain the best results when using linear kernel in SVM for signal and context features, on the other hand, we obtain the best results when using RBF kernel in SVM for structure and similarity features.

If we were to make a decision by employing a majority voting scheme with respect to the

following rules:

- If the number of the number of promoter labels is greater than the number of non promoter labels, sequence is labeled as 1.
- If the number of non promoter labels is greater than the number of promoter labels, sequence is labeled as 2.
- When the number of promoter labels equals to the number of non promoter labels, their probabilities are compared in the same manner.

Then, we would obtain 48802 true predictions and 13090 false predictions with an accuracy of 0.79 by using majority voting. In the following, we show that by integrating the component classifiers' results using a multilayer percepron, the overall accuracy is remarkably increased.

In order to evaluate Multilayer Perceptron model, five fold cross validation is used via multilayer function in the Weka [20]. For training and testing, learning rate, momentum rate, number of approaches, and number nodes in hidden layers are set to 0.3, 0.2, 500, and (input nodes + labels)/2 respectively. A grid search is performed to find the best parameters. As a result of the calculations, we found that true positive rate is 0.996, false positive rate is 0.001, specificity is 0.996, and f-measure is 0.996. The accuracy of the system is 0.997. If support vector machines were used to combine feature vectors instead of multilayer perceptron, the accuracy of the system would be 0.972. Therefore, multilayer perceptron provides better accuracy than support vector machine. These results show that selection of the model and features is successful and features represent the dataset very well. Furthermore, if the similarity feature was not added, f-measure of the system would be 0.972. The addition of the similarity features increases f-measure from 0.972 to 0.996.

4.4.2 Comparison with Existing Methods

Table 4.11 shows comparison of signal, context, structure, and similarity features with 3S1C. When signal, context, structure, and similarity features are compared individually, accuracy of context and signal features are nearly the same, and similarity feature has the lowest accuracy. On the other hand, integration of these features achieves a significantly increased accuracy.

Table 4.11: Comparison of the signal, context, structure and similarity features

Features	Models	Accuracy
Signal	SVM	0.7906
Context	SVM	0.7932
Structure	SVM	0.7747
Similarity	SVM	0.7519
3S1C	MLP	0.9970

SCS [10], which is a popular method currently, integrates signal, context and structure features by using decision trees. We obtain the sequences used in 3S1C from SCS [10] but the number of the sequences which is used in training and testing is different because of the size limitation of SVM. For fair comparison with SCS, the same training and testing data should be used. 3S1C may not be as good as these results on an other dataset. Because of the size limitation and not obtaining test data used in SCS, we can not compare our results with SCS. SCS used 30,964 promoter, 75,437 exon, 53,682 intron, and 80,538 3′ UTR sequences with 251 bps in length. The sum of sensitivity and specificity is calculated as 1.72 in SCS. On the other hand, the sum of sensitivity and specificity is calculated as 1.994 in 3S1C. If the classifier separates all promoters from non promoters, then the sum of sensitivity and specificity is 2.000. When other methods such as FirstEF, McPromoter compared to SCS , it is shown that SCS provides better results [10].

CHAPTER 5

CONCLUSION AND FUTURE WORK

Computational promoter prediction has been studied a lot in recent years. Using machine learning techniques such as discriminant analysis, Hidden Markov Models, and Artificial Neural Networks with signal, context, and structure features of DNA molecule has lead to improvements in accuracy.

Most of the promoter prediction programs and tools based on these techniques are difficult to train because they require a large amount of high-quality training data, preferably from an experimental setting. However, for most of the new genome projects there is only a limited amount of such data available. Another problem is that the outcome of programs based on these techniques is often difficult to interpret. Furthermore, all available programs are species specific; i.e., they are trained on one species and are able to predict promoters only for that particular species. Some of the promoter prediction programs only use any of the signal, context, or structure features. On the other hand, there exists many promoter prediction programs which integrate different features to predict promoter regions.

In this study, we developed promoter prediction algorithms using machine learning techniques. ProK-means and ProSVM were constructed by using structure feature of DNA sequences and results were compared with another promoter prediction method, ProSOM. We showed that ProSVM with RBF kernel when using trinucleotide conversion gives better results than ProK-means and ProSOM algorithms.

Then, 3S1C algorithm were introduced by combining signal, context, structure, and similarity features of DNA sequences. The advantage of 3S1C comes from the integration of the discriminative abilities of the signal, context, structure and similarity features that characterize promoters. Furthermore, we introduced similarity feature, and then, we showed that similarity

feature increases overall accuracy. The results of 3S1C algorithm are very promising.

As future work, we shall apply 3S1C system to other species. We shall also apply 3S1C methods to whole genome sequences. Furthermore, we shall find proximal promoters and extract promoters about 500 bps, so we may focus some distal promoters. Finally, we shall look at the differences between promoter sequence features of organisms.

REFERENCES

[1] *Wikipedia, The Free Encyclopedia* www.wikipedia.org, last accessed on 17/06/2011.

[2] Abeel, T., Saeys, Y., Bonnet, E., Rouze, P., and Peer, Y., "Generic eukaryotic core promoter prediction using structural features of DNA", *Genome Research*, vol. 18, no. 2., pp. 310-323, 2008.

[3] Zeng, J., Zhu, S., and Yan, H., "Towards accurate human promoter recognition: a review of currently used sequence features and classification methods", *Briefing in Bioinformatics*, vol. 10, pp. 498-508, 2009.

[4] Bielsa, F., C., "Using AI techniques to determine promoter location based on DNA structure calculations", Master Thesis, Universitat Politecnica de Catalunya, 2008.

[5] Abeel, T., Sayens Y., Rouze, P., and Peer, Y., "ProSOM: core promoter prediction based on unsupervised clustering of DNA physical profiles", *ISMB 2008*, vol. 24, pp. i24-i31, 2008.

[6] Perier, C., R., Praz, V., Junier T., Bonnard, C. and Buche, P., "The Eukaryotic Promoter Database(EPD)", *Nucleic Acids Research*, vol. 28, pp. 302-303, no. 1, 2000.

[7] Kanz, C., Aldebert, P. and Althorpe, N. *et.al.*, "The EMBL Nucleotide Sequence Database", *Nucleic Acids Research*, Vol. 33, Database issue, pp. D29-D33, 2005.

[8] Hubbard, T., Andrews, D., and Caccamo, M., "Ensembl 2005", *Nucleic Acids Research*, Vol. 33, Database issue, pp. D447-D453, 2005.

[9] Birney, E. *et.al*, "Ensembl 2004", *Nucleic Acids Research*, vol. 32, pp.D468-D470, 2004.

[10] J. Zeng, X. Zhao, X. Cao, and H. Yan, "SCS: Signal, Context and Structure Features for Genome-Wide Human Promoter Recognition", *IEEE Transaction on Computational Biology and Bioinformatics* , vol. 7, no. 3, 2010.

[11] Yamashita, R. et al. , "DBTSS: Database of Human Transcription Start Sites, Progress Report 2006", *Nucleic Acids Research*, vol. 34, pp. 86-89, 2006.

[12] Yamashita, R, et al., "Collection and Analysis of Eukaryotic Promoter Regions:DBTSS (DataBase of Transcriptional Start Sites)", *Genome Informatics*, vol. 13, pp. 295-296, 2002.

[13] Smale, S., Kadonaga, J., "The RNA polymerase II core promoter", *Annu Rev Biochem*, vol 72, pp. 449-79, 2003.

[14] Wang, J., et al. "MetaProm: a neural network based meta-predictor for alternative human promoter prediction",*BMC Genomics*, 8:374, 2007.

[15] Zhao, X., Xuan, Z., Zhang, M., Q., "Boosting with stumps for predicting transcription start sites", *Genome Biol*, 8:R17, 2007.

[16] Goni, J., R., Perez, A., Torrents, D., and Orozc, M., "Determining promoter location based on DNA structure first-principles calculations", *Genome Biology*, 8:R263, 2007.

[17] Rani, T., S., Durga, B., S., Bapi, R., S., "Promoter Recognition using dinucleotide Features : A Case Study for E.Coli", *9th International Conference on Information Technology (ICIT'06)*, 2006.

[18] Xie, X., Wu, S., and Lam K., M. *et al.* "PromoterExplorer: an effective promoter identification method based on the adaBoost algorithm", *Bioinformatics* , vol. 22, no. 22, pp. 2722-2728, 2006.

[19] Zhao, X., Xuan, Z., and Zhang, M., Q., "Boosting with stumps for predicting transcription start sites", *Genome Biology*, 8:R17, 2007.

[20] *Weka Toolbox* www.cs.waikato.ac.nz/ml/weka/, last accessed on 17/06/2011.

[21] Won, H., H. et al. "EnsemPro: an ensemble approach to predicting transcription start sites in human genomic DNA sequences", *Genomics 91*, pp. 259-266, 2008.

[22] Wasserman, W., W., Palumbo, M., and Thompson, W. *et.al.*, "Human-mouse genome comparisons to locate regulatory sites", *Nature Genetics*, vol. 26, 2000.

[23] *DBTSS Database* http://dbtss.hgc.jp/, last accessed on 25/05/2011.

[24] Baldi, P., Chauvin, Y., and Brunak, S., "Computational Applications of DNA Structral Scales", *ISMB 98*, AAAI, 1998.

[25] Brukner, I. *et al.*, "Sequence dependent bending propensity of DNA as revealed by DNase I: parameters for trinucleotides", *The EMBO Journal*, vol. 14, no 8, pp. 1812-1818, 1995.

[26] Dai, Q,, Liu, X., Q., Wang, T., M., "Numerical Characterization of DNA Sequences Based on the k-step Markov Chain Transition Probability", *Wiley InterScience*, 2006.

[27] Chang, C., "LIBSVM: a library for support vector machines", www.csie.ntu.edu.tw/ cjlin/libsvm, last accessed on 27/05/2011.

[28] *SVM^{light} Toolbox* http://svmlight.joachims.org/, last accessed on 17/06/2011.

[29] Drucker, H., Wu, D., and Vapnik, V., "Support Vector Machines for Spam Categorization", *Transactions on Neural Networks*, vol.10, pp. 1048-1054, 1999.

[30] Campbell, C., "Algorithmic approaches to training support vector machines", 2000.

[31] Bishop, C., M., "Pattern Recognition and Machine Learning", 2006, Singapore.

[32] Smola, A and Scholkopf, B, "A tutorial on support vector regression", tech. rep., NeuroCOLT2, 1998.

[33] Joachims, T., "A statistical learning model of text classification with support vector machines", *The Conference on Research and Development in Information Retrieval (SIGIR)*, ACM, 2001.

[34] Duda, O., R., Hart, P.E., and Stork, G., D., "Patter Classification", *Second Edition*, 2001, Canada.

[35] *Multilayer Perceptron Tutorial* www.realintelligence.net, last accessed on 15/06/2011.

[36] Gardner, M.W. and Dorling, S. R., "Artificial Neural networks (The Multilayer Perceptron)- A Review of Applications in the Atmospheric Sciences", *Atmospheric Environment*, vol. 32, no.14/15, pp. 2627-2636, 1998.

[37] K. Hornik, M. Stinchcombe and H. White, "Multilayer feedforward networks are universal approximators", *Neural Networks*, 2, 359-366.

[38] Belhadj, T., Yahia, N., B. and Zghal, A., "Automation of the Choice of the Welding Processes by a Neuronal Approach", *International Journal of Advanced Intelligence*, vol. 2, pp.57-71, July, 2010.

[39] Yegnanarayana, B., "Artifical Neural Networks", 2006, New Delhi.

APPENDIX A

SUPPORT VECTOR MACHINE AND MULTILAYER PERCEPTRON

In this appendix, detailed information about Support Vector Machines (SVMs) and Multilayer Perceptron (MLP) methods is given. These methods are used in Chapter 3.

A.1 SUPPORT VECTOR MACHINES

Support Vector Machines (SVMs), introduced by Vladimir Vapnik [29], are spatial kernel-based methods that have sparse solutions. Kernel function is evaluated on only small part of data points and prediction of new inputs depends only on this kernel function. Therefore, SVM runs faster than other memorable methods. SVM is a popular method for solving classification and regression problems [30].

SVM has some advantages among other classification methods: it constructs an optimal separating hyperplane by solving a quadratic programming problem. It separates the data by finding a hyperplane which maximizes the margin between the two classes.

In this study, support vector machines are used for classification problems with two classes. SVMs are based on Lagrange Multipliers. Therefore, firstly, the Lagrange Multiplier is reviewed and then support vector classification is explained in detail.

A.1.1 Lagrange Multipliers

This part includes limited information about Lagrange multipliers and adapted from [31]. Lagrange multipliers are used to find the local maximum or minimum points of a function of

several variables subject to one or more constraints.

To find the maximum of a function *f(x)* subject to $g(x) = 0$, the Lagrangian function is defined by A.1 and then the maximum point of $L(x, \lambda)$ is found with respect to both x and λ.

$$L(x, \lambda) = f(x) + \lambda.g(\mathrm{x}) \tag{A.1}$$

where $\lambda \neq 0$ is known as a Lagrange multiplier. Note that λ can have either sign. The maximum point is found by setting $\nabla_x L = 0$. For an N-dimensional vector x, Lagrange multiplier method gives N+1 equations that determine both local maximum or minimum points and the value of λ. A simple example is given in the following in order to illustrate this technique.

Suppose that a function $f(x_1, x_2) = x_1^2 + 2x_2^2 + 5$ is given and it is desired to find maximum point of this function subjected to constraint $g(x_1, x_2) = x_1 - x_2 + 3 = 0$. The Lagrange function for this problem is given by A.2

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}_1, \mathbf{x}_2) + \lambda.g(\mathbf{x}_1, \mathbf{x}_2) \quad \Rightarrow \quad L(\mathbf{x}, \lambda) = \mathbf{x}_1^2 + 2.\mathbf{x}_2^2 + 5 + \lambda.\mathbf{x}_1 - \lambda.\mathbf{x}_2 + 3.\lambda \tag{A.2}$$

The goal is to find the maximum points of f function with respect to the constraint. For this purpose, it is differentiated with respect to x_1, x_2 and λ respectively as shown in A.3, A.4 and A.5. If these equations are solved, the maximum point is found as $(x_1, x_2) = (0, 3)$.

$$2.\mathbf{x}_1 + \lambda = 0 \tag{A.3}$$

$$4.\mathbf{x}_2 - \lambda = 0 \tag{A.4}$$

$$\mathbf{x}_1 - \mathbf{x}_2 + 3 = 0 \tag{A.5}$$

The problem of maximizing a function subject to an equality constraint has been considered up to now. If the problem of maximizing *f(x)* subjected to an inequality constraint of form $g(x) \geqslant 0$ is considered, then the Lagrange function is given by Equation A.6 :

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda.g(\mathbf{x}) \tag{A.6}$$

Maximum point is found by taking the derivative with respective to λ and x in Equation A.6. Separating hyperplane should satisfy the conditions in Equation A.7, A.8 and A.9 according to the Karush-Kuhn-Tucker (KKT) theorem [32].

$$g(\mathbf{x}) \geqslant 0 \tag{A.7}$$

$$\lambda \geqslant 0 \tag{A.8}$$

$$\lambda.g(\mathbf{x}) \geqslant 0 \tag{A.9}$$

If there are several constraints, the problem converts to

$$maximize\ \ f(x)\ \ subject\ \ to\ \ g_j(x) = 0\ \ for\ \ j = 1, ..., J\ \ and\ \ h_k(x) \geqslant 0\ \ for\ \ k = 1, ..., K.$$

Lagrange function is given for above problem by Equation A.10

$$L(\mathbf{x}, \{\lambda_i\}, \{\mu_k\}) = f(\mathbf{x}) - \sum_{j=1}^{J} \lambda_j.g_j(\mathbf{x}) - \sum_{k=1}^{K} \mu_k.h_k(\mathbf{x}) \tag{A.10}$$

Subject to $\mu_k \geqslant 0$ and $\mu_k.h_k(x) = 0$ for k = 1, 2, ... K, where $\{\lambda_i\}$ and $\{\mu_k\}$ are Lagrange multipliers.

A.1.2 Support Vector Classification

SVM is a powerful classification technique in linear and non parametric classification. Binary classification case may be considered to understand the support vector classification technique as illustrated in Figure A.1.

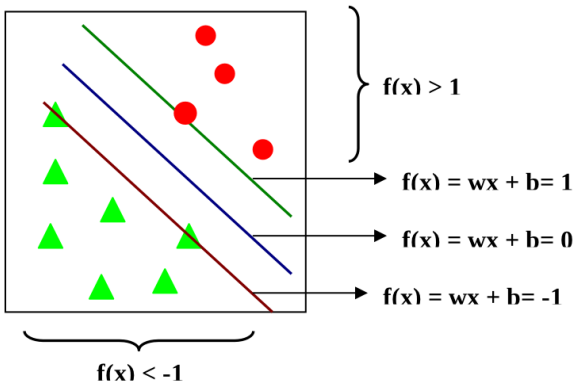

Figure A.1: Data separation by maximizing margin

Looking at the figure, there is an input vector "x" and a weight vector "w". Each data point has a label $y = \{-1, +1\}$. The decision function is $f(x) = sgn(w.x + b)$ where b is bias [33]. If data points are separable, it can be written that

x_i, $y_i(w.x_i + b) \geqslant 0$ for all x_i.

The reason is explained in the following proposition.

- $w.x + b > 0 \;\; \Rightarrow \;\; y_i.(w.x + b) \geqslant 0$ for all $y_i \geqslant 0$
- $w.x + b < 0 \;\; \Rightarrow \;\; y_i.(w.x + b) \geqslant 0$ for all $y_i < 0$

The goal of SVM is to find a function which can accurately separate the feature spaces by minimizing the error function. To achieve this goal, it obtains a hyperplane which divides data points into two groups while maximizing the margin. The margin is the maximum distance between the hyperplane and the closest data points to the hyperplane and calculated by A.11.

$$\frac{y.f(x)}{||w||} = \frac{y.(w.x + b)}{||w||} \tag{A.11}$$

Thus, problem is converted to find a maximum margin by solving equation A.12:

$$\max_{w,b}\{\frac{1}{||w||}\min_{i}\{y_i.(w.x_i + b)\}\} \tag{A.12}$$

This problem is difficult to solve. To simplify this problem, it is converted to an equivalent problem by making rescaling $w \rightarrow K.w$ and $b \rightarrow K.b$ such that equation A.13 holds for the points that are closest to the surface which is defined as support vector, so support vectors hold equation A.13 [31].

$$y.f(x) = y.(w.x + b) = 1 \tag{A.13}$$

In this case, all data points will satisfy the constraints $y.(w.x + b) \geq 1$. Therefore, the problem is converted to equation given by A.14.

$$h(w) = \frac{1}{2}.||w||^2 \tag{A.14}$$

$subject\ \ to\ \ y_i(w.x + b) \geqslant 1$

The significant reason for taking square of w in equation is to make derivative with respect to w meaningful. This problem can be solved using Lagrange multiplier given by Equation A.15.

$$L = L(x, b, \alpha) = \frac{1}{2}.||w||^2 - \sum_{k=1}^{K} \alpha_k.(y_i.(x_i + b) - 1) \tag{A.15}$$

The Lagrangian function, L should be minimized by taking partial derivatives with respect to w and b. By setting the derivatives with respect to w and b, Equations A.16 and A.17 are obtained respectively.

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{N} \alpha_i.y_i.x_i = 0 \Rightarrow w = \sum_{i=1}^{N} \alpha_i.y_i.x_i \tag{A.16}$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{N} \alpha_i.y_i = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i.y_i = 0 \tag{A.17}$$

Finally, by using Karush-Kuhn-Tucker conditions [32], Equation A.18 is obtained.

$$\alpha_i.(y_i.(x_i.w + b) - 1) = 0 \tag{A.18}$$

If w calculated by Equation A.16 is written in Equation A.15, dual Lagrange function is obtained in Equation A.24. Equation A.19, A.20, A.21, and A.22 show the interval steps of calculating dual form of Lagrangian function L.

$$L = \frac{1}{2}.\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i.(y_i.(x_i.w + b) - 1) \tag{A.19}$$

$$L = \frac{1}{2}.w.w^T - \sum_{i=1}^{N} \alpha_i.(y_i.(x_i.w + b) - 1) \tag{A.20}$$

$$L = \frac{1}{2}.\sum_{i=1}^{N} \alpha_i.y_i.x_i.\sum_{j=1}^{N} \alpha_j.y_j.x_j - \{\sum_{i=1}^{N} \alpha_i(y_i.(x_i.\sum_{j=i}^{N})\alpha_m.y_m.x_m + b) - 1)\} \tag{A.21}$$

$$L = \frac{1}{2}.\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i.\alpha_j.x_i.x_j.y_i.y_j - \sum_{i=1}^{N}\sum_{j=1}^{N})\alpha_i.\alpha_j.x_i.x_j.y_i.y_j + \sum_{i=1}^{N} \alpha_i \tag{A.22}$$

$$L = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}.\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i.\alpha_j.x_i.x_j.y_i.y_j \tag{A.23}$$

Then the problem becomes

$$\textit{minimizing}\ \ L = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}.\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i.\alpha_j.x_i.x_j.y_i.y_j \tag{A.24}$$

subject to $\sum_{i=1}^{N} \alpha_i.y_i = 0$ and $\alpha_i \geq 0$

The problem given in Equation A.24 is easier to solve then the equivalent problem of Equation A.15 because the dual form does not include w and b parameters.

A.1.3 Non Linear Separable SVM

If the data is not linearly separable, it is converted into a high dimensional Hilbert Space. This conversion is made by using non linear kernel functions. Figure A.2 illustrates this conversion. For finding the optimal separating hyperplane, the "kernel trick" is used. The

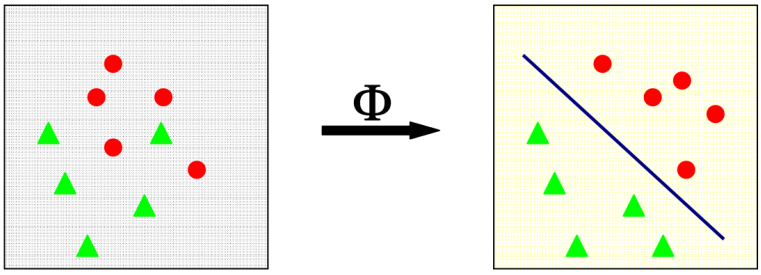

Figure A.2: Data conversion from Eucledian space to Hilbert space

inner product in Lagrange function can be replaced by kernel function in higher dimensional Hilbert space [34].

$$x_i.x_j \Longrightarrow \Phi(x_i).\Phi(x_j) \tag{A.25}$$

The kernel function is defined by Equation A.26

$$K(x, y) = \Phi(x)^T.\Phi(y) \tag{A.26}$$

The most commonly used kernel functions are listed below :

- $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)$ (Linear)
- $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i \cdot \mathbf{x}_j + r)$ (Sigmoid)
- $K(\mathbf{x}_i, \mathbf{x}_j) = e^{\frac{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}}$ (RBF)
- $K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i \cdot \mathbf{x}_j + r)^d$ (Polynomial)

where $\gamma > 0, r$, and d are the kernel parameters. Sometimes we might prefer a solution that better separates the data while ignoring a few noisy data. In such cases, slack variables $\xi_i > 0$ are introduced where $\xi_i > 1$ will be misclassify [31]. Then we write our problem :

$$minimizing \ \frac{1}{2}.w.w + c.\sum_{i=1}^{N} \xi_i \tag{A.27}$$

subject to $y_i.(w.x_i + b) > 1 - \xi_i, \xi_i > 0$

If we write the Lagrangian according to slack variables

$$L(w, b, a) = \frac{1}{2}.\|\mathbf{w}\|^2 + c.\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}a_n.(y_n.f(x_n) - 1 + \xi_n) - \sum_{i=1}^{N}\beta_n.\xi_n \tag{A.28}$$

where $\{\alpha_n \geq 0\}$ and $\{\beta_n \geq 0\}$

The corresponding set of KKT conditions are given by

$$\alpha_n \geq 0 \tag{A.29}$$

$$y_n.f(x_n) - 1 + \xi_n \geq 0 \tag{A.30}$$

$$\alpha_n.(y_n.f(x_n) - 1 + \xi_n) = 0 \tag{A.31}$$

$$\beta_n \geq 0 \tag{A.32}$$

$$\xi_n \geq 0 \tag{A.33}$$

$$\alpha_n.\beta_n = 0 \tag{A.34}$$

where n=1,2...,N and w, b, and (ξ_n) parameters are found by taking derivatives of the Lagrange function with respect to w, b, (ξ_n) respectively.

A.2 MULTILAYER PERCEPTRON

Multilayer perceptrons (MLP) sometimes called Artifical Neural Networks (ANN) were first introduced by McCulloch and Pitts with the introduction of perceptron which is a one layer neural network [35].

In 1958, Rosenbert generated the perceptron model and this model includes three layers [36]:

- a “retina” which sends inputs to the following layer
- “association unit” which compounds the inputs with weights
- “the output layer” that compounds the values

Multilayer perceptron were developed to model the human brain which is formed by neurons which consist of billions of individual cells. Multilayer perceptron is a kind of neural network as illustrated in Figure A.3.

A MLP includes an input layer, hidden layers, and an output layer. Figure A.4 shows a multilayer perceptron with one input, 2 hidden layers, and 1 output layer.

MLP can model data by using usually non-linear functions sometimes called activation function without making prior assumption about distribution of the data. Commonly used activation functions are:

- the threshold funtion: if x is positive, 0 ; otherwise 1
- the sigmoid function: $1/(1 + e^{-x})$
- the hyperbolic tangent function: $\{e^{2x} - 1\}/\{e^{2x} + 1\}$

Multilayer perceptron classifies data by using supervised learning, so training needs dataset with output vectors. Multilayer perceptron can learn through training and weights are regulated up to developing of satisfaction input - output mapping [36, 37]. Various algorithms may be used for the training step of multilayer perceptron. Steps of the algorithm are shown in Figure A.5. LSM standing for Least Square error may be replaced with any kind of error function in the figure.

A.2.1 The Backpropagation Algorithm

One of the most popular methods for training multilayer perceptron is the backpropagation algorithm which is similar to a gradient descent method. The back propagation algorithm involves the following steps [36]:

Algorithm 1 Backpropagation Algorithm

1: Initialize weight vectors and set node offsets as desired.
2: Present first input node from training data to the network.
3: Propagate the input node over the network to acquire an output.
4: Compute the error function by matching real output with the desired output.
5: Propagate the error function again over the network
6: Regulate weights by reducing total error.
7: Iterate steps 2-7 with the next input node upto the desired total error.

Detailed information about multilayer perceptron can be found in [36] and [31].

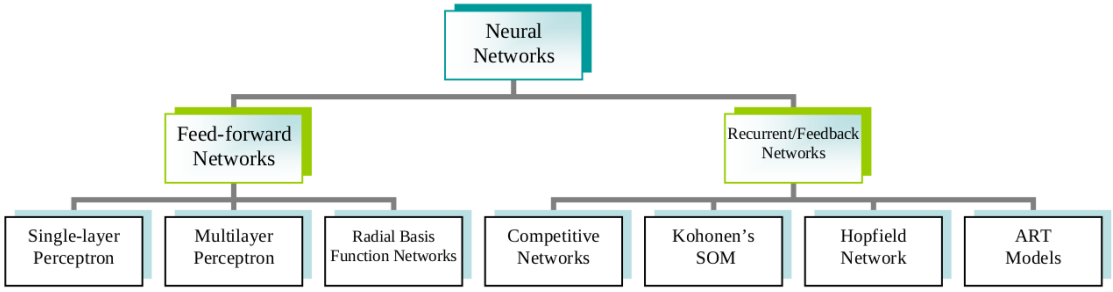

Figure A.3: Schematic representation of Neural network

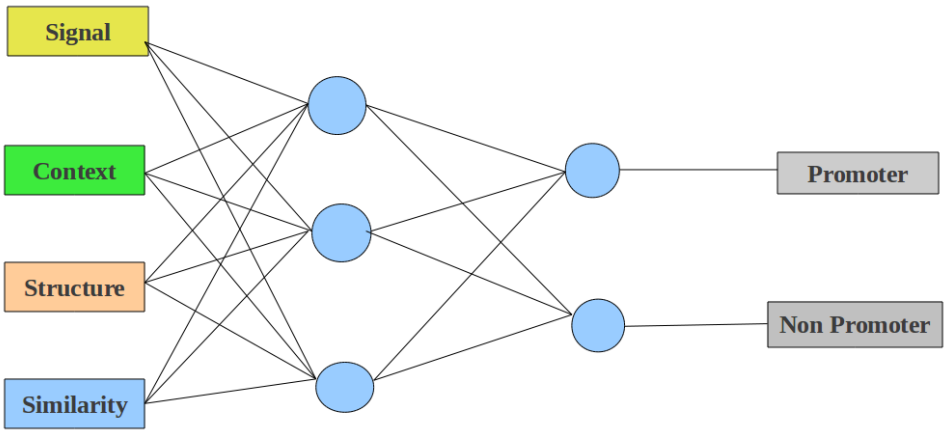

Figure A.4: Multilayer perceptron with one input, two hidden and one output layer

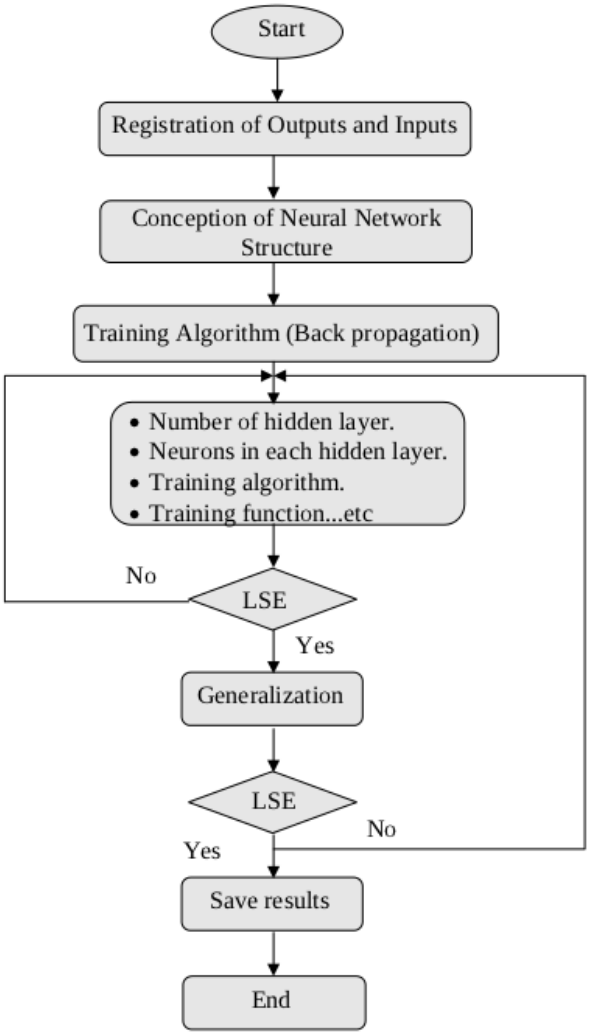

Figure A.5: Steps of multilayer perceptron algorithm (Adapted from [38])

Some characteristic of the multilayer perceptron makes it a state of the art method. They are listed in the following:

- Multilayer Perceptron does not make any prior assumption about distribution of the data.
- There is no need for decisions to relative importance of the several inputs measurements during training phase [36].
- The network performs routinely many operations in parallel and also a given task in a distributed manner [39].
- It is flexible that is, the network adjusts to a new environment easily.

APPENDIX B

PROK-MEANS RESULTS

In this appendix, complete results related to ProK-means algorithm are given. Firstly, we run ProK-means algorithm and obtain k clusters. Secondly, we determine cluster probabilities. Then, we choose the highest cluster probability as the threshold and we calculate precision, recall, and f-measure values. Also, we choose second highest cluster probability as the threshold and we repeat these calculations until obtaining the highest f-measure value. Table B.1, B.2, B.3, and B.4 show dinucleotide validation results for k=16, 25, 36, and 49, respectively and Table B.5, B.6, B.7, and B.8 show trinucleotide results for k=16, 25, 36, and 49, respectively.

Table B.1: Dinucleotide conversion validation results for k=16

Threshold	Precision	Recall	F-measure
0.932	1.000	0.375	0.545
0.746	0.939	0.596	0.729
0.400	0.841	0.663	0.742
0.379	0.762	0.894	**0.823**
0.221	0.686	0.904	0.780

Table B.2: Dinucleotide conversion validation results for k=25

Threshold	Precision	Recall	F-measure
1.000	1.000	0.010	0.019
0.959	1.000	0.202	0.336
0.896	1.000	0.365	0.535
0.809	0.965	0.529	0.683
0.733	0.951	0.558	0.703
0.584	0.914	0.712	0.800
0.500	0.890	0.779	**0.831**
0.350	0.848	0.808	0.828
0.292	0.789	0.827	0.808

Table B.3: Dinucleotide conversion validation results for k=36

Threshold	Precision	Recall	F-measure
1.000	1.000	0.019	0.038
0.950	1.000	0.240	0.388
0.930	1.000	0.337	0.504
0.880	1.000	0.394	0.566
0.860	1.000	0.413	0.585
0.840	0.982	0.538	0.696
0.770	0.969	0.606	0.746
0.750	0.956	0.625	0.756
0.480	0.854	0.731	0.788
0.340	0.809	0.856	**0.832**

Table B.4: Dinucleotide conversion validation results for k=49

Threshold	Precision	Recall	F-measure
1.000	1.000	0.058	0.109
0.986	1.000	0.240	0.388
0.923	0.972	0.337	0.500
0.875	0.974	0.365	0.531
0.857	0.962	0.490	0.650
0.844	0.947	0.519	0.671
0.808	0.955	0.606	0.741
0.630	0.957	0.644	0.770
0.625	0.944	0.654	0.773
0.585	0.878	0.760	0.814
0.419	0.832	0.760	0.794
0.375	0.827	0.779	0.802
0.372	0.817	0.817	**0.817**

Table B.5: Trinucleotide conversion validation results for k=16

Threshold	Precision	Recall	F-measure
0.614	0.577	0.394	0.469
0.485	0.602	0.740	**0.664**
0.320	0.559	0.769	0.648
0.254	0.513	0.779	0.618
0.230	0.488	0.798	0.606

Table B.6: Trinucleotide conversion validation results for k=25

Threshold	Precision	Recall	F-measure
0.635	0.543	0.240	0.333
0.498	0.602	0.538	0.569
0.473	0.613	0.731	**0.667**
0.381	0.566	0.788	0.659
0.349	0.547	0.837	0.662
0.286	0.520	0.856	0.647

Table B.7: Trinucleotide conversion validation results for k=36

Threshold	Precision	Recall	F-measure
0.690	0.690	0.279	0.397
0.590	0.638	0.490	0.554
0.570	0.589	0.538	0.563
0.530	0.592	0.587	0.589
0.500	0.564	0.596	0.579
0.390	0.561	0.663	0.608
0.330	0.511	0.663	0.577
0.320	0.521	0.817	**0.637**
0.310	0.512	0.817	0.630
0.300	0.489	0.817	0.612

Table B.8: Trinucleotide conversion validation results for k=49

Threshold	Precision	Recall	F-measure
0.857	0.000	0.000	0.000
0.750	0.000	0.000	0.000
0.687	0.545	0.115	0.190
0.667	0.623	0.317	0.420
0.521	0.614	0.413	0.494
0.509	0.626	0.548	0.585
0.486	0.619	0.577	0.597
0.420	0.608	0.596	**0.602**
0.414	0.606	0.589	0.597
0.365	0.584	0.595	0.589

Figure B.1, B.2, B.3, and B.4 show the relationship between f-measure and threshold for dinucleotide conversion, and Figure B.5, B.6, B.7, and B.8 show the relationship between f-measure and threshold for trinucleotide conversion.

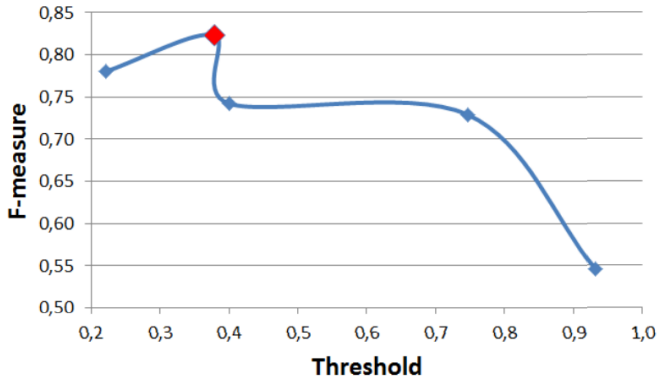

Figure B.1: DiCon: Relationship between threshold and f-measure for k=16

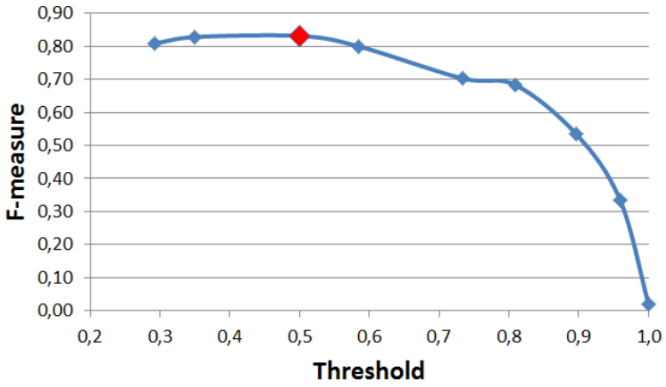

Figure B.2: DiCon: Relationship between threshold and f-measure for k=25

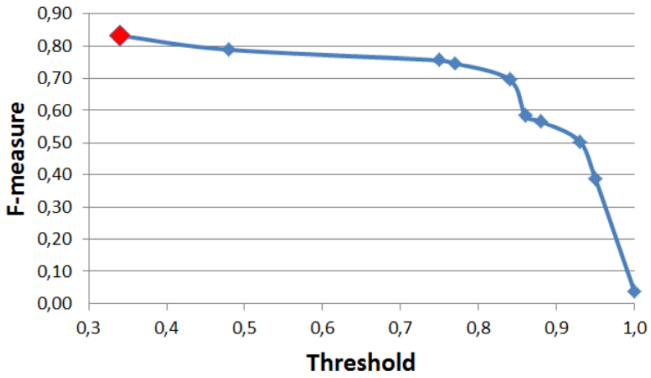

Figure B.3: DiCon: Relationship between threshold and f-measure for k=36

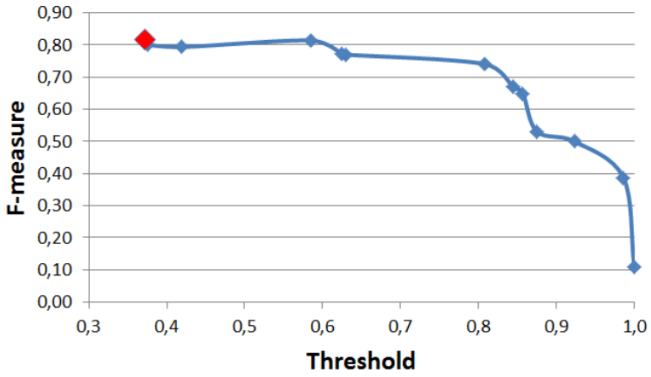

Figure B.4: DiCon: Relationship between threshold and f-measure for k=49

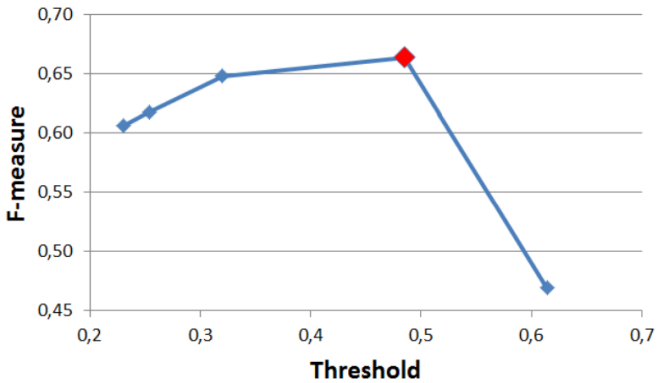

Figure B.5: TriCon: Relationship between threshold and f-measure for k=16

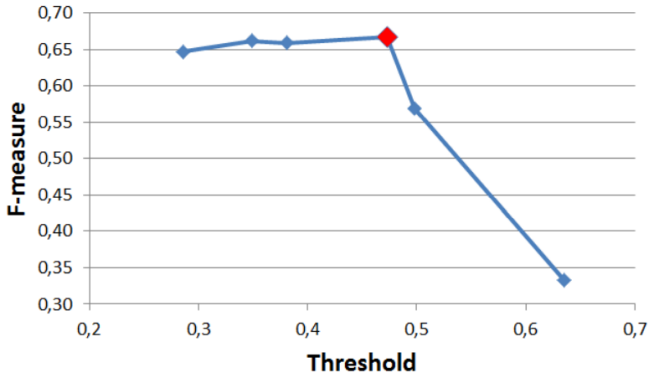

Figure B.6: TriCon: Relationship between threshold and f-measure for k=25

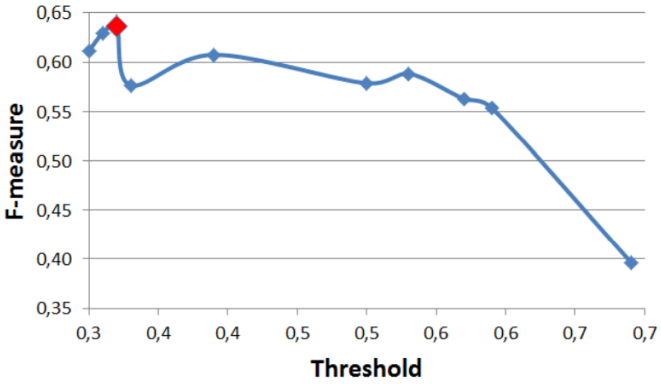

Figure B.7: TriCon: Relationship between threshold and f-measure for k=36

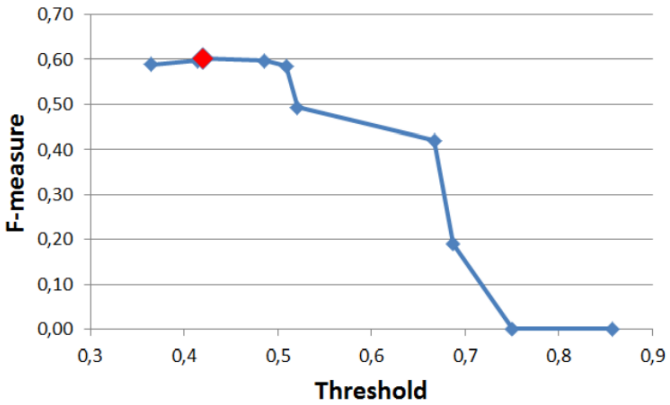

Figure B.8: TriCon: Relationship between threshold and f-measure for k=49

Figure B.9, B.10, B.11, and B.12 show the relationship between precision and recall values for dinucleotide conversion.

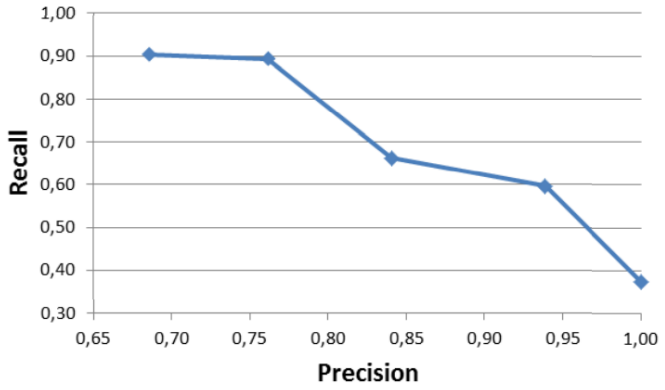

Figure B.9: DiCon: Relationship between precision and recall for k=16

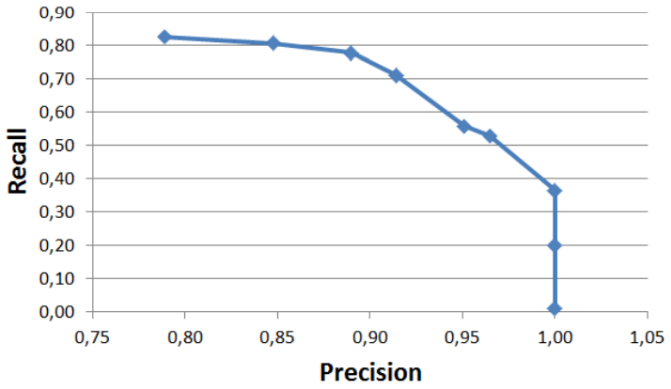

Figure B.10: DiCon: Relationship between precision and recall for k=25

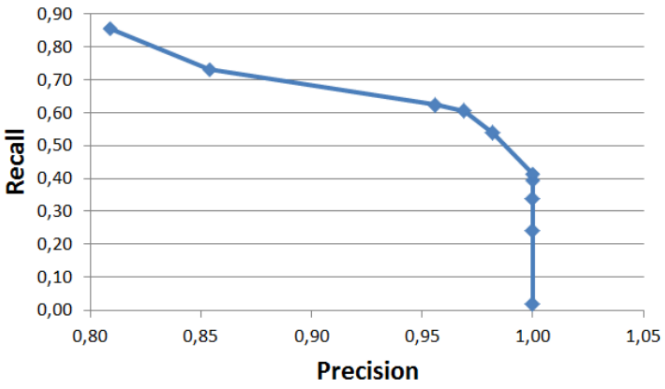

Figure B.11: DiCon: Relationship between precision and recall for k=36

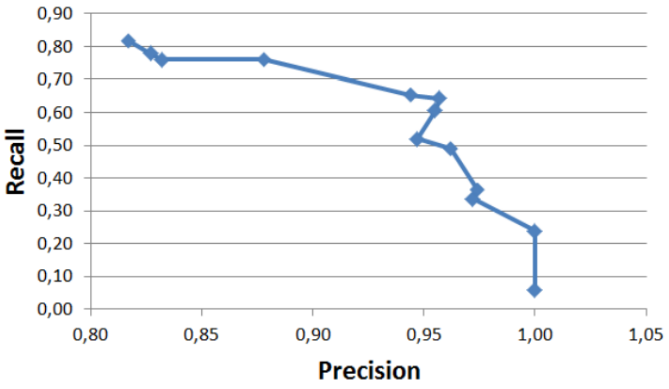

Figure B.12: DiCon: Relationship between precision and recall for k=49

Figure B.13, B.14, B.15, and B.16 show the relationship between precision and recall values for trinucleotide conversion.

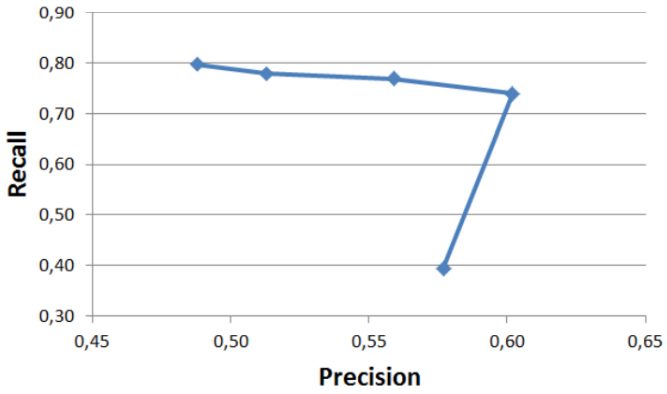

Figure B.13: TriCon: Relationship between precision and recall for k=16

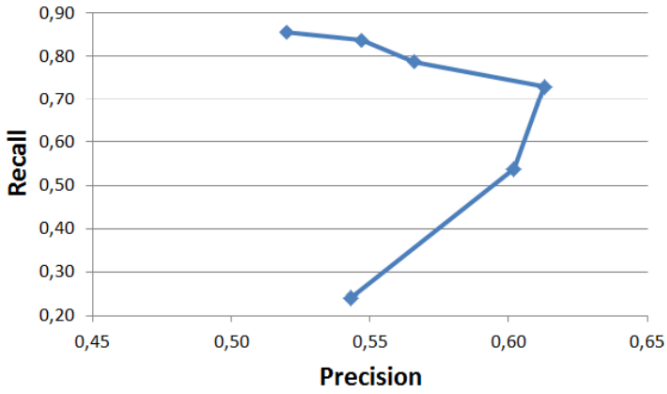

Figure B.14: TriCon: Relationship between precision and recall for k=25

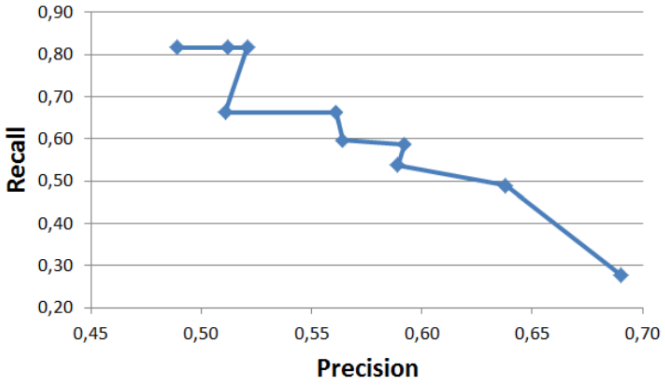

Figure B.15: TriCon: Relationship between precision and recall for k=36

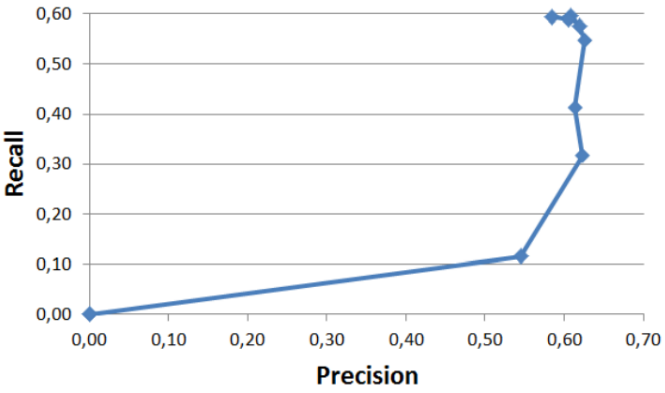

Figure B.16: TriCon: Relationship between precision and recall for k=49

APPENDIX C

GLOSSARY OF THE TERMS

- TSS: Transcription Start Site
- bps: Base pairs
- A: Adenine
- G: Guanine
- C: Cytosine
- T: Thymine
- DNA: Deoxyribonucleic acid
- RNA: Ribonucleic acid
- EPD: Eukaryotic Promoter Database
- DBTSS: Database of Transcription Start Site
- SVM: Support Vector Machine
- MLP: Multilayer Perceptron
- DiCon: Dinucleotide conversion
- TriCon: Trinucleotide conversion
- BSE: Base Stacking Energy
- RBF: Radial Basis Function
- TP: True Positive

- TN: True Negative
- FP: False Positive
- FN: False Negative