NTNU – Trondheim
Norwegian University of
Science and Technology

RAFT

Real And False TFBSs

Sandesh Prasai

Medical Technology
Submission date: July 2014
Supervisor: Pål Sætrom, IDI
Co-supervisor: Finn Drabløs, IKM

Norwegian University of Science and Technology
Department of Computer and Information Science

RAFT-Real and False Transcription Factor Binding Sites

By: Sandesh Prasai

7/11/2014

Abstract

Transcription factors are proteins essential for regulation and expression of genes. The region of DNA where transcription factors binds during transcription is known are transcription factor binding site. The clear understanding of transcription factors and its binding sites reveals different genomic secrets. The aim of this work is to classify the real and false transcription factor binding sites using machine learning approach. Model based prediction of binding site like Position Weight Matrix (PWM) has been successfully used to identify the transcription factor binding site, but generates a lot of false positive binding sites. For this reason, this project tries to classify the real and false positive binding site based on the genomic and physical properties.

Firstly, this work studies the properties for classification of real and false binding sites. In a second stage, the true binding from ChIP-seq region was used to make positive and false positive binding regions. The genomic property and statistical measures of physical properties were computed from both regions forming positive and negative datasets. The statistical measure like standard deviation, mean, kurtosis and skewnees were computed. Finally, these properties were used to train the model and followed by testing. The result of classification was compared with three different machine learning algorithms like Support Vector Machines (SVM), Random Forest (RF) and Naïve Bayes (NB).

The results of the experiment showed that SVM are well suited to classify the real and false transcription factor binding sites. However, RF predicts with better accuracy, specificity and sensitivity in all observed cases. It was shown that Pearson VII function-based Universal Kernel (PUK) in SVM predicts with better accuracy than other kernels. It was also showed that only few attributes were important in classification. Furthermore, presence of additional signals was observed around transcription factor binding sites from correlation plot. The result of logo plot indicated that transcription factor binding sites may form a cruciform structure. But an analysis performed to verify the cruciform structure did not clearly reflect the structure.

This work demonstration that combined genomic and statistical measures of structural properties can classify real and false transcription factor binding sites. This project can be further enhanced to make a general classifier tool by identifying transcription factor independent properties.

Acknowledgements

First and foremost, I would like to thank my department IDI for giving me an opportunity to work in the topic RAFT- Real and False Transcription Factor Binding Sites. I wish to express my sincere gratitude to Professor Finn Drabløs at department of Cancer Research and Molecular Medicine (IKM), Norwegian University of Science and Technology (NTNU), for his close guidance and readiness to help during my work. This work would not have been possible without his constant guidance and valuable suggestions during the project and report writing. His vast and never ending insight in the topic had been a great source of motivation and inspiration for this work.

I would also like to thank my department Professor Pål Sætrum for his help and cooperation. Last but not least, I would like to thank my faculty “Faculty of Natural Sciences and Technology”, IDI and IKM departments for their close cooperation and coordination for providing me a working environment to accomplish this project.

Sandesh Prasai

Preface

This text is submitted as partial fulfillment of the requirements of degree in MSc in Medical Technology, specialization in Bioinformatics at the Norwegian University of Science and Technology (NTNU). This project has been carried out at the department of Cancer Research and Molecular Medicine (IKM) during the period of February 2014 to July 2014 under the guidance and supervision of Professor Finn Drabløs and Pål Sætrum. I would like to thank both of them for their continuous guidance and feedback during my work. Their valuable comments have helped me a lot to accomplish this project with fruitful output.

Trondheim, Norway, July 2014

Sandesh Prasai

Contents

List of Figures

List of Tables

Abbreviations

A	Adenine
T	Thymine
C	Cytosine
G	Guanine
RAFT	Real And False Transcription Factors
BLAST	Basic Local Alignment Search Tool
BED	Browser Extensible Data
bp	Base-Pairs
ChIP-seq	Chromatin Immunoprecipitation Sequencing
DNA	Deoxyribonucleic acid
GTF	Gene Transfer Format
MCC	Matthew Correlation Coefficient
nts	Unit-Nucleotides
PWM	Position Weight Matrix
PSSM	Position Specific Scoring Matrix
PFM	Position Frequency Matrix
RNA	Ribonucleic Acid
SELEX	Systematic Evolution of Ligands by Exponential Enrichment
SNPs	Single Nucleotide Polymorphisms
SVM	Support Vector Machine
SMOTE	Synthetic Minority Oversampling Technique
TF	Transcription Factor
TSS	Transcription Start Site
TFBS	Transcription Factor Binding Site
UCSC	University of California Santa Cruz
USF1	Upstream Stimulatory Factor 1
PUK	Pearson VII function-based Universal Kernel
RF	Random Forest

NB	Naïve Bayes
SD	Standard Deviation
PCR	Polymerase Chain Reaction
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
SRY	SEX-determining RegionY
HSF	Heat Shock Factor
HIFs	Hypoxia-Inducible Factors
QP	Quadratic Programming
SMO	Sequential Minimal Optimization
GUI	Graphical User Interface
FIMO	Find Individual Motif Occurrences
SOAP	Simple Object Access Protocol
WEKA	Waikato Environment for Knowledge Analysis
NCBI	National Center for Biotechnology Information

Chapter 1

1 Introduction

Transcription is the process of synthesizing a sequence of Ribonucleic Acid (RNA) from a complementary template strand of Deoxyribonucleic acid (DNA). Proteins that monitor the turning on and off of genes in the genome are called transcription factors (TF). These proteins are essential for regulation and expression of genes. Understanding such complex interaction will reveal the fact of responding the cell to various environments. TFs are also responsible for the cell reaction to extracellular information. In addition, TFs are important regulatory parameters as they are responsible for deciding the fate of individual cells. TF may bind directly to promoter regions of DNA or directly to the RNA polymerase molecule. The region where these TFs bind is known as Transcription Factor Binding Sites (TFBSs).

1.1 Problem statement

The location of TFBSs is crucial in deciphering the fundamental cellular process like growth control, cellular hormone secretion, cell-cell communication etc.[1] There are various in vitro and in vivo throughput experiments and techniques to find binding sites. DNA microarray based techniques like Chromatin Immunoprecipitation (ChIP-ChIP) is in-vivo high throughput techniques to predict TFBS. Similarly, SELEX (Systematic Evolution of Ligands by Exponential Enrichment) is an in-vitro high throughput technique to predict high affinity binding sites. SELEX involves iterative steps of processes like separating aptamers from non-aptamers discriminating target-bound DNA from Free DNA and amplification of these obtained target-bound DNA by polymerase chain reaction (PCR) [2] . These processes make SELEX a slow and resource demanding. Recently, ChIP-seq has become a popular and powerful tool to determine TFBSs at a genome-wide scale [3] . However, the availability of ChIP-quality antibodies against each TF is the main limitations of this approach. The solution for this problem could be a computational approach for the prediction of TFBSs. There are various computational techniques to find the TF and TFBS. However, computational methods are not free from limitations. In profile type model like position weight matrix (PWM), the choice of threshold is important parameter to control false positive. Lowering the threshold will generate lots of false positive binding sites. Alternatively, strict choice of threshold may miss the actual binding sites [4] [5] . Another vital assumption made in PWM is position independent nucleotide, meaning that there is no correlation between the neighbor nucleotides [6] .

Most prediction methods for finding potential DNA binding sites for a specific transcription factor (TF) use a model for the TFBS, and compare each position of the DNA sequence (e.g. a genome) against this model. Any position with a significant score against the model may then be classified as a potential binding site. Some examples of common models are e.g. consensus sequence, hidden Markov model and in particular PWM. A PWM is a profile-type model, where each column of the matrix contains a probability (log odds ratio) of finding each base (A, C, G, and T) at that position of the motif. The log odds are summed over the PWM, and sequence positions scoring better than a chosen cutoff are used as positive TFBS predictions. The main problem with this approach is that it generates a large number of false positive TFBS predictions. It has actually been estimated that in most cases the estimate will be completely dominated by false positives [4] . One important approach for filtering out false positives has been to use cell type-specific information. In particular, DNase hypersensitivity (HS) data and histone modification data can be used to identify active regulatory regions in a given cell type. However, in addition to being cell type-specific, this approach is also often limited by lack of suitable data. Therefore, a common point of interest in Bioinformatics community is deriving any better way for distinguishing between true and false positive binding sites.

1.2 Objective of project

The prediction of accurate binding site has been a common interest of all genomic research centers. There are enduring challenges to find the precise binding site of higher eukaryotes due to large genome size and presence of introns and variable lengths of TFBSs which add extra challenge for simple experimental and prediction methods. There are lots of ongoing efforts to predict binding site based on different genomic, evolutionary, chemical and physical properties. This project “RAFT- Real And False Transcription Factor Binding Sites” is the extension of spring project “Building pipeline for genomic tracks” submitted in December 2013. RAFT tries to classify the real and false TFBSs using machine learning approach. For this approach, RAFT uses genomic properties and physical properties of dinucleotide as input vector for classifier. To build a property based genomic track, RAFT is based on assumption that the TFBSs are depended on local parameters which reveal the property of TFBSs [7] [8].

The main objectives of the thesis are as follows:

- Get familiarisation with format and source of track
- Make a genomic track based on context based features
- Develop a robust pipelines for processing and comparison of genomic tracks
- Use machine learning approach to classify the real and false transcription factor binding sites
- Compare the result of classification with other approaches to achieve an optimal accuracy

1.3 Approach

In this project, machine learning method was implemented for the prediction of true and false positive transcription factor binding sites. It was focused to develop a property based robust genomic track for identification of real binding sites for a given TF, independent of cell type. The basic assumption was that real TFBSs are found in a suitable genomic context, whereas random binding sites will lack any common context. A suitable context is mainly associated with the properties of regulatory regions, as active TFBS in general will be found in such regions. Then, the idea was to use properties those are associated with regulatory regions to develop a classifier for PWM-based TFBS predictions. To handle the classification task, I employed support vector machines (SVMs), which are the form of supervised machine learning approach. The TFBS mapped computationally for a ChIP-seq data was used as original data for calculation of property based input vectors. I also tried to compare the result obtained from SVM with results obtained from other algorithms like random forest (RF) and Naïve Bayes (NB).

1.4 Organization of thesis

In this dissertation, I classified real and false TFBSs using machine-learning technique. I used genomic features and statistical measures of physical properties to train the classifier. In Chapter 1, I have introduced the problem and the objective of this work. The concise approach to solve the problem stated is also shown in the same chapter. The background theory, related work and the choice of features are presented in Chapter 2. In this chapter, some prediction and classification method using SVM that are closely related to this dissertation are presented. In Chapter 3, the methodology to accomplish the project is presented. It also describes the work flow diagram of this project and the quality measure taken into account to assure the confidence of prediction. The result and discussion of the work is shown in Chapter 4. The results obtained with different kernel function on same dataset are presented. Finally, the conclusion of dissertation and further enhancement possibility are presented in Chapter 5. Last but not least, references and appendices to this work are also presented in the end of report.

Chapter 2

2 Background and Literature Review

Transcription factor (TF) is a protein that plays a vital role in regulating gene expression in living organism. This proteins bind physically to their target loci is a key step of activating or repressing a gene. There are approximately 1700-1900 TFs in human that binds in different DNA segments like promoters, enhancers, silencer, insulators and other control regions [9] [10] . Prediction of transcription factor binding sites (TFBSs) is vital but is extremely challenging problem due to the large genomic size, the short and variable length of binding sites. In addition, transcription factors target vary between the different types of tissues, stage of development and physiological conditions [3] . These dynamic regulations make more complication in finding the proper binding regions. Some binding sites are located close to Transcription Start Site (TSS) like in promoter and some are located very far like in enhancer for instance; 1 mega base pair far from target gene in eukaryotes [11] . However, different bioinformatics research institutions have shown their interests to reveal the secret behind TFBSs with computational as well as biological approaches.

2.1 Significance of TFBS

In this section some key roles of TFBSs are explained.

Development

Some of the TFs are involved in the development of organism. For example, TFs encoded by SEX-determining RegionY (SRY) protein is responsible for the initiation of male sex determination in human [12] .

Response to environment

TFs are responsible to provide adaptability to organism with the environment. For example TF like heat shock factor (HSF) is responsible to regulate gene necessary to adapt in higher temperature environment [13] . Similarly, Hypoxia-inducible factors (HIFs) helps to survive cell in low oxygen environments by up regulating genes [14] .

Cell cycle control

Transcription factors are responsible to control cell divisions, the shape and size of cells [15] . For instance, C-MYc codes for the transcription factor whose mutation version is found in may cancers [16] .

Pathogenesis

To prevent from pathogens, transcription factor can alter gene expression in host cell that will boost the defense mechanism of the host cell.

The interaction between three dimensional structure of DNA and TF to recognize the binding site is based on local geometry of base pairs [17] . There are four standard structural motifs that TF recognize as binding sites. These structures are helix-turn-helix (HTH), Zinc Finger (ZF), Basic Leucine Zipper (B-ZIP), Basic Helix-Loop-Helix (B-HLH) [18] . In some cases, proteins can directly recognize the special structure of DNA like cruciform and bind DNA hairpins [18] .

2.2 Cruciform structure of DNA

Cruciform structures also known as hairpin are important regulators of biological processes [19] [20] . The structures of cruciform comprises of a steam, a branch point and a loop as shown in Figure 2-1. The size of the loop in cruciform depends on the length of the gap between inverted repeats. Generally, AT-rich gap sequences increase the probability of cruciform formation in DNA. The study has identified two classes of cruciform structure [21] . The first class is unfolded, and has square planer structure. It is characterized by 4-fold symmetry in which adjacent arms are almost perpendicular to one another. Another class of cruciform has folded conformation. In this class, adjacent arms form an acute angle with the DNA strands as shown in Figure 2-2. The detection of cruciform conformation was first described in circular plasmid DNA. The structure was stabilized by negative superhelix density [22] . This type of structure is vital since the distortion on this type of structure results in the failure or reduction in replication [23] .

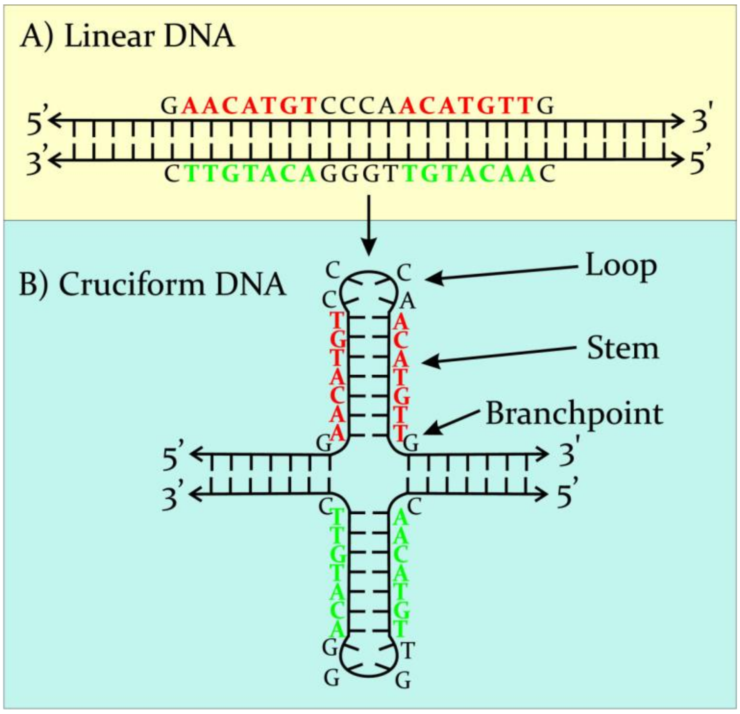


Figure 2-1: Example of cruciform structure

Figure shows a cruciform structure as linear DNA (A) and as an inverted repeat (B), taken from [24]

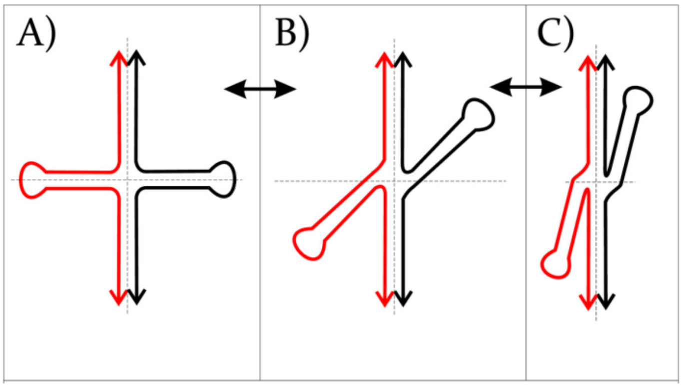


Figure 2-2: Conformations of cruciform

Figure shows a three different conformations of cruciform structure; (A) unfolded with 4 fold symmetry, (B) bent and (C) stacked with 4 chains of DNA in close vicinity, taken from [24]

To predict the cruciform structure computationally for a sequence, a reverse complement of that sequence is computed. The basic idea is to calculate the similarity between sequence and reverse compliment of it. Similarity matrix can be viewed as dot matrix plot where the value in the matrix increases by 1 if the similarity is found between position i and j. A high score value across the diagonal is expected for cruciform structure. It can be explained in following steps;

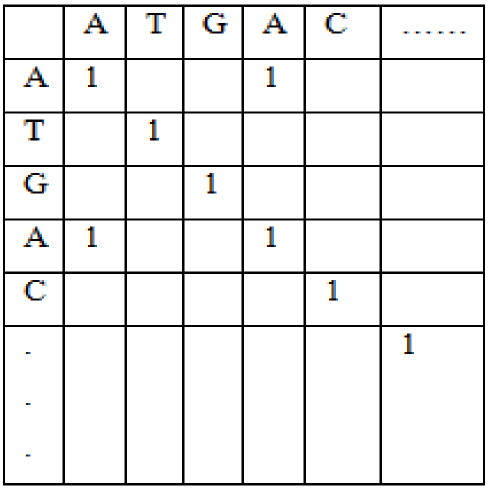
- For a given sequence find the reverse complement of that sequence
- Make a matrix M of size NxN with sequence in row and its reverse complement in column
- Position M(i, j) of this matrix corresponds to a possible interaction between position i and position j of a sequences
- If the residues at i and j matches, then add 1 to that position in the matrix
- Do this over all pairs of positions in the sequence
- Loop for overall sequences
- If there is a tendency for cruciforms, then this matrix should show an increased score along a diagonal

These steps can be explained with following small example. Let us consider a small given sequence ATGACTTGATTCAAGTCAT to test the cruciform structure. Then, the reverse complement of given sequence will be ATGACTTGAATCAAGTCAT. The similarity computation between these two sequences is shown in Table 2-1 shows a similarity matrix computation for one single sequence and its reverse complement to verify the cruciform structure. The first base in row and column is A therefore, the value of cell is 1. In similar way, the score is computed for all the bases and this is repeated until all the sequences are finished. The final score of matrix will be used to predict the cruciform nature of DNA.

This matrix can be characterized with a pattern of high scores along its diagonal.

Table 2-1 shows a similarity matrix computation for one single sequence and its reverse complement to verify the cruciform structure. The first base in row and column is A therefore, the value of cell is 1. In similar way, the score is computed for all the bases and this is repeated until all the sequences are finished. The final score of matrix will be used to predict the cruciform nature of DNA.

Table 2-1: Similarity matrix



	A	T	G	A	C	......
A	1			1		
T		1				
G			1			
A	1			1		
C					1	
- - -						1

In study carried out by Brázda et al. in [24] , proteins that identify cruciform and interact with it are classified into 4 different families.

Junction resolving enzymes

These proteins are found in several organisms like bacteria, yeast, archaea and mammals. It will bind to junctions of any sequence. Examples of these proteins are RuvC, Cce1, Ydc2, Integrases, RusA etc.

Proteins involved in transcription and DNA repair

This family of proteins is involved in mechanism like DNA repair which is key mechanism for genomic stability. The DNA binding proteins like BARCA1, polymerase 1, Rad54, Hop1, P53 binds to cruciform structure. Interestingly, some proteins can also stimulate the formation of cruciform after binding with DNA [25] .

Chromatin-associated proteins

This family of protein are found in the cell nucleus and involves in different mechanisms like modulating chromatin structure, remodeling of DNA topology etc. some proteins like DEK and BARCA1 are involved in DNA replication and repair. They play an important role in maintaining genomic stability as they are able to diffuse the stress generated during transcription and replication.

Proteins involved in replication

Cruciform structure indicates as recognition signal near eukaryotic origins of DNA replication. There are many proteins that bind to cruciform structure during replication. S16, AF10 are DNA binding proteins that structure-specific.

From all above discussions, it is known that for genomic stability, holiday junction and long cruciform structure is necessary. Deregulation of these proteins may lead to deletions, carcinogenesis, DNA translocation and loss of genomic stability that may be lethal. Therefore, mutation, epigenetic modification, single nucleotide polymorphisms and insertion in cruciform structure can destroy the cellular process.

2.3 Approaches for prediction of TFBS

The prediction of TFBS in eukaryotes is extremely difficult. However, there are many approaches used to predict the binding site. Some of the approaches are based on genomic features, physical properties, evolutionary conservation and binding motif distribution. The main approaches for prediction of binding sites can be classified into two methods like experimental and computational method. Some of these sections had been discussed in report of spring project.

2.3.1 Experimental method

There are some high throughput in vitro and in vivo methods that experimentally verify TFBS. SELEX an in vitro method for finding target ligands was developed 20 years ago. SELEX identify a small numbers of aptamer from original library that binds with high affinity to a protein of interest [26] . These aptamers are oligonucleotide ligand with length of 15-60 bases [28] . The steps involved in SELEX are as follows: First of all the target molecule is defined. Then a library of oligonucleotide is created which is very large in numbers (10^{15}) [29] . The oligonucleotide binding the target molecule is amplified by PCR. Then, the few oligonucleotides will bind to target which are called aptamers. The oligonucleotides that are not binding are separated from those forming aptamers. This will decrease the number of high affinity binding molecules from large number to few set. These aptamers are then amplified with PCR. Finally, each aptamers are isolated, sequenced and refined.

Similarly, another method used Chip coupled with DNA microarray to predict the binding site for given protein. This method is called ChIP on ChIP, which allows predicting the entire spectrum of in vitro DNA-binding sites of that protein [30] . Most recently, “ChIP-seq” a microarray based in-vivo technology is commonly used to determine binding sites. The ChIP-seq work flow is shown in Figure 2-3.

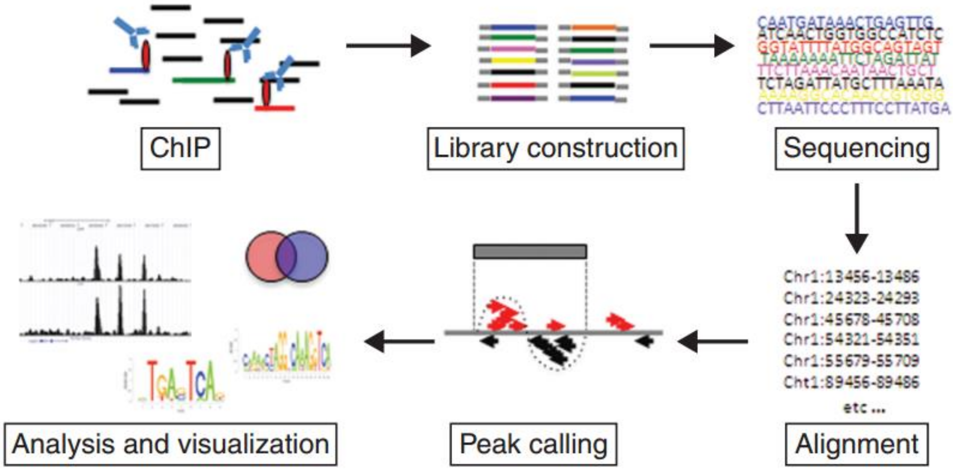


Figure 2-3: Steps involve in ChIP-seq procedure

Figure shows from ChIP procedure to sequencing procedure, finally showing the binding region under the peak, taken from [31]

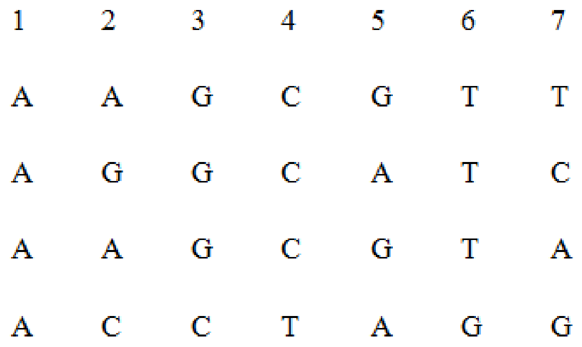
The first step in ChIP-seq is to crosslink DNA with DNA-binding proteins. The chromatin is fragmented into pieces of 150 to 500 bp using sonication. After the fragmentation is done, the immunoprecipitation is done using the specific antibody against the protein of interest. The quality of antibody has great impact in the result. It will produce a millions of short fragments directional DNA tags. The lengths of these tags are 35-50 bp. This is ready for sequencing, where these small fragments of variable lengths tags are aligned to a reference genome of the sample organism. This step is known as peak calling [27] .

2.3.2 Computational method

Computational prediction of TF binding sites (TFBSs) is based on position weight matrix (PWM) also called position-specific scoring matrix (PSSM). PWM scores binding motifs based on the observed nucleotide patterns in a set of TFBSs for the representing TF. It is produced on the basis of gapless local multiple alignment of sequences. PSSM scores correspond to the conservation of residue at that position. The motif is scanned in all sequences in different file format like FASTA. A sliding window method is used to search the motif and score is computed from the segment of equal length as that of motif. If the score is found higher than the threshold then the motif is the binding site. If not, it will continue checking till the end of the sequence.

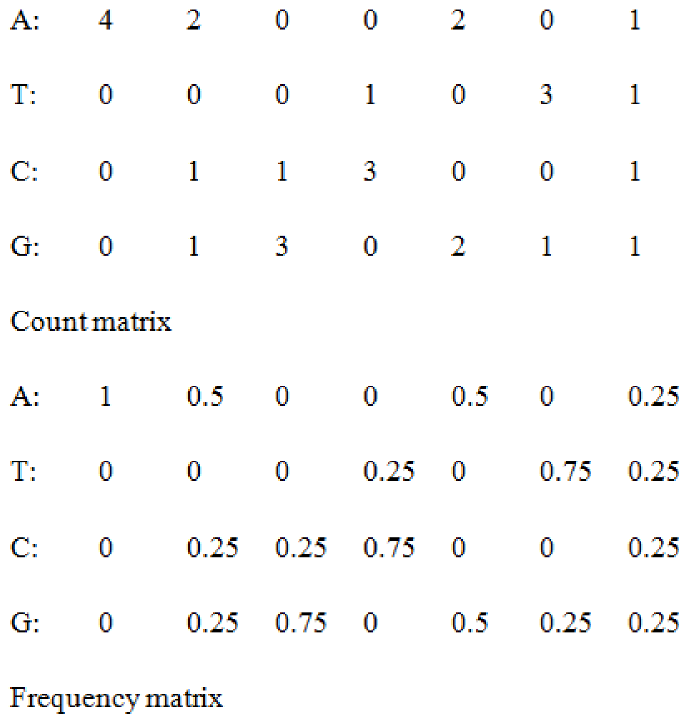
2.3.2.1 Construction of PWM

During a search, initial similarity search of a query against the sequence in database is done. The hits are used to make a multiple alignment and build a frequency matrix by counting the occurrence of each nucleotide at each position of alignment. This matrix has four rows (A, G, C and T) but the columns are equal to the length of motif is called Position Frequency Matrix (PFM). PWM is now calculated from PFM. This is demonstrated as an example below. Let us consider four sequences of length 7. Then the sequences are aligned first in first step. The second step is followed by counting the number of bases to construct PFM.



1	2	3	4	5	6	7
A	A	G	C	G	T	T
A	G	G	C	A	T	C
A	A	G	C	G	T	A
A	C	C	T	A	G	G

Figure 2-4: Aligned sequences



A:	4	2	0	0	2	0	1
T:	0	0	0	1	0	3	1
C:	0	1	1	3	0	0	1
G:	0	1	3	0	2	1	1

Count matrix

A:	1	0.5	0	0	0.5	0	0.25
T:	0	0	0	0.25	0	0.75	0.25
C:	0	0.25	0.25	0.75	0	0	0.25
G:	0	0.25	0.75	0	0.5	0.25	0.25

Frequency matrix

Figure 2-5: Count and frequency matrix

Let us suppose, we have a multiple alignment of N sequences. Let, $n_{u,b}$ is the number of residue of type b present at column u. then, fraction of residue of type b at position u is given by $f_{u,b}$,

$f_{u,b} = \frac{n_{u,b}}{N}$………… (1)

Let $m_{u,a}$ be the score of residue “a” at position “u”, then

$m_{u,a} = \sum f_{u,b} S_{a,b}$ ………. (2), where $S_{a,b}$ is score matrix element

In logarithmic form

$m_{u,a} = \sum \frac{\ln(1-\bar{f}_{u,b}}{\ln(N/(N+1))} S_{a,b}$…… (3), where $\bar{f}_{u,b} = \frac{n_{u,b}}{N+1}$, giving extra weight to conserved position

Then, $m_{u,a} = log \frac{q_{u,a}}{p_a}$ ……. (4) is the log odd ratio that gives the probability of a occurring at u and the probability of a occurring by random chance. A small value called pseudo count is usually added to prevent the case of log (0) and to prevent the loss of small score.

This resulting scoring matrix can be used to search the entire input DNA sequences in order to find regions similar to the original set of known regions. A cutoff or threshold value is used to ensure the match between input sequence and motif. PWMs have been used for the prediction of the binding affinity for numerous bacterial and eukaryotic TFs. It used in database like TRANSFAC and JASPAR for this searching purpose [32] .

2.3.2.2 Limitations of PWM

Though PWM is very useful in locating TFBS, it has potential drawbacks. PWM do not account for flexible length motifs. Secondly, PWM assumes each nucleotide participates independently in the corresponding DNA-protein interaction, meaning that there is no presence any correlation among the nucleotides in different alignment column [33] . This assumption is the main reason to generate large false positive predictions.

2.3.2.3 Tradeoff between cutoff and false positive

The false positive prediction can be controlled by selection of cutoff or threshold. However, it again has a drawback. The requirement of rigorous match (higher cutoff) will increase the sensitivity and is likely to result in fewer false positive predictions but can potentially result in more sites being missed (false negative) [4] [50] . Adversely, lowering the PWM score threshold will increase the number of false-positive hits.

2.3.2.4 Enhance prediction accuracy with PWM

There are many approaches that have been used to improve the performance of PWM. Dinucleotide PWM is a simple extension of PWM that outperform classical PWM. In [33] , Kulakovskiy et al. developed a tool called DIChipMunk. In this tool, they account the effect (correlation) of neighboring nucleotide in input sequences converting mononucleotide to dinucleotide. This method is handy than PWM for very large training set of sequences. In addition, this method also outperforms chipMunk that perform better than PWM. However, the computation slower than PWM since dinucleotide has more parameters to compute.

Some methods use additional information like co-localization and conservation of TFBSs to improve the prediction accuracy. There are some methods that use motif documented in database like JASPER and TRANSFAC. Comet [34] , Cluster-Buster [35] and ModuleMiner [36] are examples of such methods. Other methods use clustering information in addition to motif conservation to increase prediction accuracy. Examples of such types of method are Stubb and ELL [37] . Another approach to increase the prediction is by reducing the false positive count, which can be done by filtering the false positive values, classifying the false positive from true positive value. There are several approaches that used machine-learning technique to increase the prediction.

In RAFT, statistical measures of physical properties of DNA and genomic features are combined as input vector to classifier. It is entirely a novel concept than that has been attempted by other researchers. Most of the researches have been carried out with the properties like evolutionary distance, sequence profile, k-mers, charge, hydrogen bonding, hydrophobicity etc. However, the attempt to classification with genomic distance like GC skew, distance from TSS to binding region, distance from CpG to binding region has not been attempted. In the same way, physical properties and physiochemical properties has been used in other methods. However, statistical measures like mean, standard deviation, skewness and kurtosis of physical properties have been used in this project that is a unique approach. Some of the closely related researches using machine-learning approach are discussed in next section.

2.3.3 Related work

SVM has been used widely in computation biology from long time for task like prediction of disease risk [38] classification of genomic and proteomic data [39] , cancer classification [40] microarray data classification [41] etc. There are many algorithms based on different approaches for prediction and classification of TFBS. Most of them used genomic, physiochemical and evolutionary properties for prediction and classification purpose, some of them are discussed in this section.

In [42] , Nassif et al. attempted to predict the protein-glucose binding site using SVM. They used Random Forest method to find the key features and then used as the descriptor for classifier. The features included in this method were physio-chemical properties like charge, hydrogen bonding and hydrophobicity. They used 29 protein-glucose binding sites for training purpose and 14 for testing. Negative dataset with three groups of sites was taken that does not bind glucose. Non-sugar binding, sugar binding and non-binding were label used for three groups of sites. Leave one out cross validation method was used to cross check the output of classifier along with holdout independent testing set. The train-to-test ratio of 2:1 was used. The specificity and sensitivity rate of combined features were found as 93.33% and 89.66% respectively.

The prediction of RNA binding sites in a protein using SVM and PSSM profile was carried in [43] by Kumar et al. For this purpose they train two models using amino acid sequence and evolutionary information. They used 86 RNA binding protein chains and evaluated using 5 fold cross validation technique. For the first model, fixed pattern was generated from RNA interacting chains, if the central residue was found to be interacting residue then the pattern was assigned as positive else the pattern was assigned as negative. Each amino acid was represented by vector of 21 including one dummy amino acid. The accuracy of 76.05% and Mathew's Correlation Coefficient (MCC) of 0.31 was achieved. In second model, evolutionary information was obtained from PSSM generated during PSI-BLAST search. The accuracy of 81.16% and the MCC of 0.45 were obtained by PSSM approach, which was significantly higher than that of using evolutionary information using single sequence. The webserver 'Pprint' using this algorithm to predict RNA binding residue can be found in http://www.imtech.res.in/raghava/pprint/. Their SVM model based upon amino acid sequences preforms slightly better than existing technologies but model based on evolutionary information developed from PSSM outperforms all existing and even ANN model developed by Jeong and Miyano in 20006 [43].

In [44] , Holloway et al. attempt to integrate eight different types of features to predict TFBS in Saccharomyces Cerevisiae genome. Binding site degeneracy, conservation measures, clusters, TF target correlation, target-target correlation, GO annotation, phylogenetic profiles, k-mer distribution were the eight genomic data used as property for classification purpose. The positive examples were taken from ChIP-ChIP and other experiments and negative data were randomly chosen from the output of MotifScanner that does not show motif for particular TF. This model achieved good sensitivity and specificity and thus able to detect false positive binding sites.

In [45] , an attempt to predict binding site in the mouse genome using support vector machines was done by Sun et al. The data consist of merger of promoters for mouse annotated with TFBS from databases like ABS and ORegAnno. The problem with the imbalance of dataset was handled using Synthetic Minority Oversampling Technique (SMTOF) that will increase the number of minority class elements. The data includes 250 upstream, non-coding sequence and background data. Background dataset are negative dataset that were drawn 5000-4500 base pair away from any gene. They used 47 annotated promoter sequences in total. Sequences extracted from ABS database were 500 bp in length and those extracted from PRegAnn were 2000 bp long. The model was trained with SVM and the result of prediction and performance of classifier was found better than other prediction algorithms like MotifLocator and EvoSelex.

In [46] , Mukherjee et al. aims to predict binding site on helix turn helix type of transcription factors in eukaryotes. For this purpose, they used 90 sequences of transcription factor with helix turn helix retrieved from NCBI for training set. Four properties used to classify these transcription factors were evolutionary sequence conservation, positively charged residues, hydrogen bond donor-acceptor and hydrophobic residues. Evolutionary conservation of residue was obtained from multiple sequence analysis. Positively charges residues are found to be more affinity with negatively charged residues of DNA strand. SVM with RBF kernel was used to train and classify the model. Using k-fold cross resampling technique for cross validation, the output of classifier was very impressive giving the accuracy of 94.19%, sensitivity of 96.7% and specificity of 89.16%.

In another approach [47] , Maienschein-Cline et al. uses physiochemical features of DNA to predict TFBS. These features were derived from Gibbs energy of amino acid interactions and DNA structure. Since, there exist a structural correlation between the free and bound TFBSs; this property was used for training support vectors. In addition to the geometry structure, the structural profile based on hydroxyl radial cleavage of DNA was also used. In addition, a chemical feature like electrostatic profile around DNA was used. The positive training set includes TFBS sequences form RegulonDB and flanking nucleotides. Randomly selected non-coding sequences of E.Coli genome was used as negative training set. The equal length distributions of positive and negative dataset were taken so that feature dimension would be equal for both datasets. SVM with non-linear kernel was used to train and test the model. The accuracy of each variant method was accessed by cross validation method.

2.3.4 ChIP-seq as source of data

ChIP-seq data is the important source of TFBSs. This method follows two steps ChIP and sequencing. ChIP-seq is a high throughput technique used to determine in vivo binding affinities of transcription factors to DNA [48] .The peak-finder algorithm is used to process the data obtained from ChIP-seq experiment. This peak finder will locate the DNA segments containing binding signal. Figure 2-6 shows an overview of ChIP workflow, the detail steps were explained in previous section. The DNA segments thus obtained are of variable length and therefore need to process further to find the actual binding site. De novo motif discovery tool is used to predict fixed length actual binding motif. Motif is the common pattern shown by the binding sites of transcription factor.

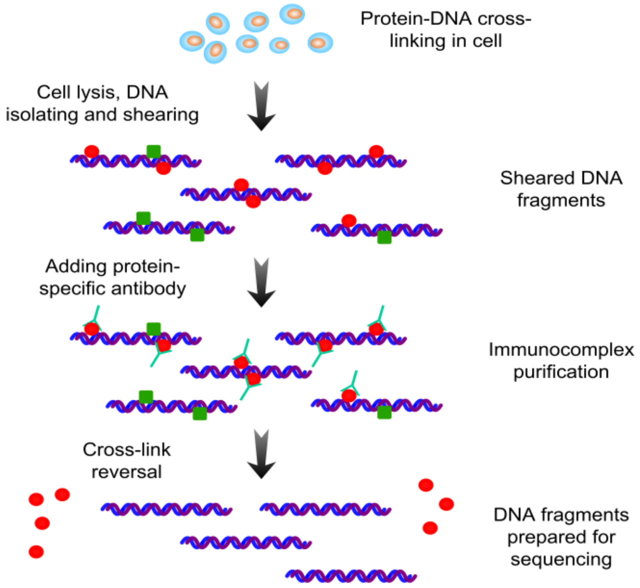

Protein-DNA cross-linking in cell
Cell lysis, DNA isolating and shearing
Sheared DNA fragments
Adding protein-specific antibody
Immunocomplex purification
Cross-link reversal
DNA fragments prepared for sequencing

Figure 2-6: ChIP-seq work flow

Figure focus in the preparation of ChIP, the output of ChIP is DNA fragment with variable lengths, taken from supplementary material of [49] .

2.3.5 Physical Properties of DNA

Statistical measures like Skewness, standard deviation, kurtosis and mean for the physical properties like stacking energy, propeller twist, protein induced deformability, duplex disrupt energy, duplex free energy, DNA denaturation, BDNA twist, protein DNA twist, stabilizing energy of ZDNA were used [5] [51] .

Stacking energy

Overall structure of double helix depends upon the sequence of base. It is calculated from quantum mechanics by applying energy required to de-stack the DNA helix. It is expressed in kilocalories per mol. High peaks in base stacking reflect regions of the helix that de-stack or melt more easily; conversely a minimal peak would represent more stable regions [52] .

Propeller twist

The dinucleotide propeller twist angle scale can be measured by X-ray crystallography of DNA oligomers. A region with high propeller twist would mean that the helix is quite rigid in this area. Correspondingly, regions that are quite flexible would have low propeller twist values. It was found that AT base pairs have higher levels of Propeller twist than that of GC base pair [53] .

Protein induced deformability

The deformability of DNA plays important role its packaging in the cell and recognition by other molecules in cellular process. This can be acquired from empirical energy function examining crystal structures of DNA–protein complexes. The larger value in the scale shows more deformable sequence and smaller value reflects less deformability of DNA helix [54] .

Duplex disrupt energy

The DNA disrupt energy was calculated using calorimetric calculation on 19 DNA oligomers and 9 DNA polymers using nearest neighbor approach. Thermodynamic data was used to calculate the stability of DNA duplex structure and found that stability was depended on base sequence but not the base composition [55] .

Duplex free energy

To calculate the Duplex free energy, positional-dependent nearest-neighbor (PDNN) model was used. This method was initially used for describing RNA/DNA duplex formation. It was shown that region with low free energy content was more stable than region with high energy content.

DNA denaturation

It is calculated by UV electronic spectroscopy under very high resolution. It was shown that DNA with low peak is more easily denaturated than region with high peak value [51] .

B-DNA twist and Protein–DNA twist

B-DNA twist is the mean twist value of angles in B-DNA. It was calculated on 38 B-DNA crystal [51].Protein –DNA twist was calculated by average distributions of the conformational parameters that can describe the DNA variability from protein-DNA complexes.

Stabilizing energy of Z-DNA

It represents the free energy value for transition from B to ZDNA for dinucleotide. DNA stretches with low energy minima form Z-DNA than high energy region.

Before using these parameters in this project, the correlation of their values was performed for each dinucleotide. The absolute sum from the correlation matrix was calculated. The top four parameters with least correlation were taken for further analysis in this project. The value of these parameters and the absolute correlation values are presented in Table 2-2: Dinucleotide property table.

Table 2-2: Dinucleotide property table

These values of dinucleotide were used by [56] , originally used by Liao [57] , most of the values are found in dinucleotide database.

Dinuc base	**Stacking energy**	**Prop twist**	**Protein deform**	**Duplex free eng**	**Duplex disrupt eng**	**DNA denat**	**BDNA twist**	**ProtDNA twist**	**Stab eng ZDNA**
AA	-5,37	-18,66	2,9	-1,2	1,9	66,51	35,5	35,1	3,9

AC	-10,51	-13,1	2,3	-1,5	1,3	108,8	33,1	31,5	4,6
AG	-6,78	-14	2,1	-1,5	1,6	85,12	30,6	31,9	3,4
AT	-6,57	-15,01	1,6	-0,9	0,9	72,29	43,2	29,3	5,9
CA	-6,57	-9,45	9,8	-1,7	1,9	64,92	37,7	37,3	1,3
CC	-8,26	-8,11	6,1	-2,3	3,1	99,31	35,3	32,9	2,4
CG	-9,69	-10,03	12,1	-2,8	3,6	88,84	31,3	36,1	0,7
CT	-6,78	-14	2,1	-1,5	1,6	85,12	30,6	31,9	3,4
GA	-9,81	-13,48	4,5	-1,5	1,6	80,03	39,6	36,3	3,4
GC	-14,59	-11,08	4	-2,3	3,1	135,83	38,4	33,6	4
GG	-8,26	-8,11	6,1	-2,3	3,1	99,31	35,3	32,9	2,4
GT	-10,51	-13,1	2,3	-1,5	1,3	108,8	33,1	31,5	4,6
TA	-3,82	-11,85	6,3	-0,9	1,5	50,11	31,6	37,8	2,5
TC	-9,81	-13,8	4,5	-1,5	1,6	80,03	39,6	36,3	3,4
TG	-6,57	-9,45	9,8	-1,7	1,9	64,92	37,7	37,3	1,3
TT	-5,37	-18,66	2,9	-1,2	1,9	66,51	35,5	35,1	3,9

Table 2-3: Absolute correlation value of dinucleotide properties

Table shows the absolute correlation value of dinucleotide properties. The complete correlation matrix is presented in Appendix I

Here the top four properties and value with lest absolute correlation is mark in red. These values were used as attribute during classification of binding sites.

Properties	**Absolute correlation value**
Stacking energy	0,440
Propeller twist	0,528
Protein induced deformation	0,524
Duplex free energy	0,629
Duplex disrupt energy	0,572
DNA denaturation	0,469
BDNA twist	0,202
Protein DNA twist	0,343
Stab energy for ZDNA	0,542

2.3.5.1 Statistical measures

2.3.5.1.1 Kurtosis

Kurtosis is defined as the measure of peakedness of distribution of real value random variable also known as fourth momentum of a distribution [58] . Kurtosis describes the shape of probability distribution and used to compare the shape of particular distribution with normal distribution. The higher kurtosis will have distinct peak near the mean and have heavy tails. Conversely, lower kurtosis has a flat top near mean. Kurtosis has been useful parameter to analyze property. In [59] , kurtosis was used to fine molecular classifiers in cancer. Depending upon the sign of kurtosis method was implemented to find the gene that describes the outlier property.

Kurtosis is given by [47], K=$\sum_{i=1}^{N}(Y_i - \bar{Y})^4/(N-1)s^4$

Where, $\bar{Y}$ is mean

s is the standard deviation and N is the number of data points

2.3.5.1.2 Skewness

Skewness gives the measure of symmetry or a lack of symmetry of the distribution of real value random variable about its mean [60] [61]. Skewness value can be negative or positive. If the value is negative then the tail of left side of probability density function (PDF) is longer or flatter than the right side. Conversely, if the skewness value is positive then the tail on the right side is longer or flatter than left side.

Skewness is given by [47], Sk=$\sum_{i=1}^{N}(Y_i - \bar{Y})^3/(N-1)s^3$

Where, $\bar{Y}$ is mean

“s” is the standard deviation and N is the number of data points

2.3.6 Genomic Properties

This section contains the description of different genomic properties used in this project for classification purpose.

2.3.6.1 CpG distance

CpG islands (or CG islands) are regions characterized with occurrence of high frequency of CG. Almost all CpG Island are sites for transcription initiation [62] . CpG distance is the length between CpG Island and binding site motif region. This data can be downloaded from UCSC table browser in bed file format. CpG distances for positive and negative set were obtained using the bed tool. This distance was one of the properties for classification.

2.3.6.2 Transcription start site distance

TSS is a location in the DNA sequence where RNA polymerase binds and start to make RNA from DNA. TSS for USF1 can be downloaded from the UCSC table browser. The nearest distance between TSS and binding motif region was obtained using bed command. This was another property used for classification.

2.3.6.3 GC content

GC content is the percentage of G-C expressed in percentage. The GC region is found higher near the transcription binding region [63] **.** In this project, the fasta regions of positive and negative datasets were used to calculate the GC content.

$$GC\ count = \frac{count\ of\ C + count\ of\ G}{total\ count\ of\ nucleotides}$$

2.3.6.4 GC Skew

GC skew gives the genomic strand asymmetry. The number of Guanine is higher in leading strand and the number of Cytosine is higher in lagging strand [64] . On other hand, the value of GC skew is positive then it corresponds to leading strand. Similarly, if the value of GC skew is negative then it corresponds to lagging strand.

$$GC\ skew = \frac{G - C}{G + C}$$

2.3.7 FASTA Format

Fasta format is a text based format which uses single letter code for the representation of nucleotides sequences or amino acid sequences [65] . This format is accepted as query sequence for tool like BLAST (Basic Local Alignment Search Tool) search purpose. Fasta format begins with a single line description, followed by lines of sequence data [65] . The word following the ">" symbol is the identifier of the sequence, and it is optional. The greater than sign (>) distinguishes from description line to sequence line. FASTA format does not allow gaps present within the representation of sequences and is recommended to use 80 characters per line. Fasta format accepts both upper case and lower case. In addition, FASTA file can also be represented without the fasta definition, which is only the bare sequence. Sample of fasta format is shown below.

```
>crab_anapl ALPHA CRYSTALLIN B CHAIN (ALPHA(B)-CRYSTALLIN).
MDITIHNPLIRRPLFSWLAPSRIFDQIFGEHLQESELLPASPSLSPFLMR
SPIFRMPSWLETGLSEMRLEKDKFSVNLDVKHFSPEELKVKVLGDMVEIH
GKHEERQDEHGFIAREFNRKYRIPADVDPLTITSSLSLDGVLTVSAPRKQ
SDVPERSIPITREEKPAIAGAQRK
```

In this project, FASTA format is used to query the binding site for potential TFBS region through FIMO.

2.3.8 WEKA-A classifier

Waikato Environment for Knowledge Analysis (WEKA) was developed by machine learning group in University of Waikato, New Zealand. It has been popular in data mining task due to portability and ease of use. It is freely downloadable and can work with large set of data. Weka is a collection of algorithms and data visualization tool. Different algorithms for classification which is categorized as bayes, function, lazy, rules trees etc. Libsvm and Sequential Minimal Optimization (SMO) are categorized under function based classifier. Similarly, algorithms like J48, random forest etc. are classified as tree based classifier.

WEKA tool implements SVM and provides the easy GUI to select the kernels and its coefficients. It implements libsvm and new algorithm called Sequential Minimal Optimization (SMO) for training support vectors [66] . In this project, SMO is used for classification purpose. SMO can work with very large quadratic programming (QP) case as it breaks down the large QP problem into series of smallest possible QP problems. Finally, these small QP problems are solved analytically. The memory required in SMO is linear with the training data. Weka accepts arff file format directly. However, a converter is also available to convert from csv to arff file format.

Weka can be accessed via both graphical user interface and command line [67] . In this project, graphical user interface (GUI) was used as interface. Explorer is the main interface for Weka. It contains six panels like pre-processing, classify, cluster, associate, select attributes and visualize. Data can be loaded from the preprocessing panel. Filter option is available to pre-process data. One can delete, randomize data using filter tool. There is also undo option that will revert the previous action in weka. Classify panel contains the option for the classification and regression. Several algorithms are available for this purpose and also provides cross validation to evaluate these algorithms. Cluster panel gives the clustering algorithms that include k-means, normal distributions with diagonal covariance matrices. Associate panel can be used to generate rules that define the relationship between groups of attributes in the dataset. The fifth panel is select attributes that is used to identify the best attributes for classification. It gives the subset of features that are highly correlated with the class. It utilizes different methods like best-first search, exhaustive search, genetic search, greedy stepwise etc. This panel has been used in this project to predict the important attributes for classification. The last panel of is visualization, that construct the plot scatter plots for all attributes pair in dataset. Individual plot can be selected and enlarged.

2.3.9 Supervise and unsupervised learning

Machine learning problem are classified into two main categories called supervise and unsupervised learning. In unsupervised learning the cases are not labeled. In this type algorithm it cannot invent what the case is, but it could be able to cluster the data into different class based upon similarity [68] . On other hand, cases are labeled in supervise learning. There is another intermediate learning type called semi-supervise learning that learns from both labelled and unlabeled cases.

2.3.10 SVM

SVM are supervised learning models in machine learning that is mostly used in classification of data, regression, and pattern reorganization. SVM approximates really well for linear and nonlinear datasets. Different kernels are used to achieve the optimum performance of classifier which is presented in result section. The algorithm for SVM is based on statistical learning theory and the Vapnik-Chervonenkis (VC) dimension which were introduced by Vladimir Vapnik and Alexey Chervonenkis [69] . Statistical learning theory is the framework in machine learning that deals with making predictions, making decisions or constructing models from a set of data [70] .

SVM are called supervised large margin classifier and the data points touching the lines are called support vectors. From [70] , the errors bounded are associated with the margin separated by the hyperplanes. Therefore, SMV tries to find the best hyperplane by maximizing the boundary of hyperplane separating the training data. SVM is widely and successfully used in various applications like pattern recognition, face detection, data classification and text classification. SVM is widely used in data mining due to kernel trick or kernel substitution. SVM approximate well enough for linearly and non-linearly separable cases.

2.3.10.1 Support Vector Machines for linear discriminants

If we consider a linearly separable data then there might exist a plane that perfectly classifies the data into two sets. There might be lot of planes that perfectly classifies the data. A case is shown in Figure 2-7: Linear discriminants. However, SVM will try to approximate the best plane to classify the data selecting the furthest plane from both the data (in this case bold blue line) since small perturbations will not cause misclassification.

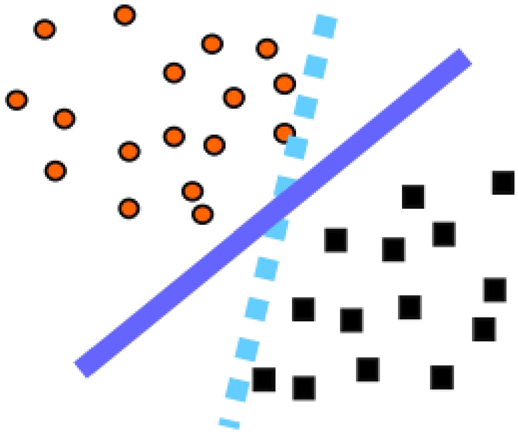


Figure 2-7: Linear discriminants

Figure shows two different datasets that can be separated by more than one linear discriminants. In this scenario a line that divides with maximam margin is used, taken from[71] .

Mathematically,

Let us consider a linearly separable data that can be classified with one or more hyperplanes shown in Figure 2-8. Suppose we have a training data D, a set of n points of the form [72] [73] ,

$$D=\left\{\left(X_i, Y_j\right)|X_i \in R^p,\ y_i \in \{-1, 1\}\right\}\ \forall\ i = 1 \text{ to } n$$

Where, Y_i is either 1 or -1, indicating the binary class to which the point X_i belongs. Each X_i is a p-dimensional real vector. We would like to find the maximum-margin hyperplane that divides the points having $Y_i = 1$ from those having $Y_i = -1$.

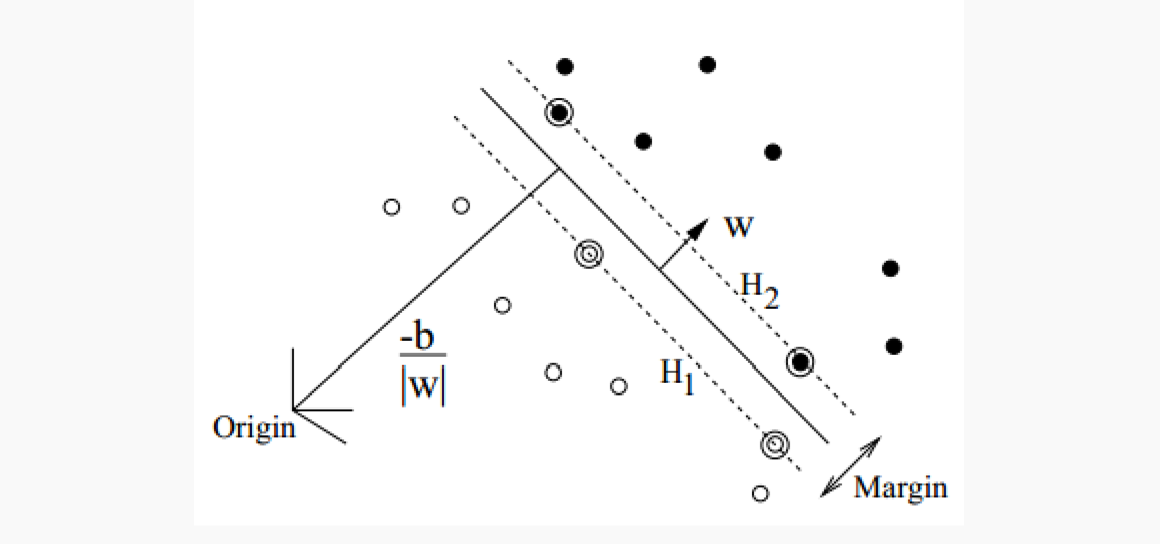


Figure 2-8: A hyperplane separating two different datasets

Figure shows the mathematical representation of linear discriminants, taken from [73] .

Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors

The data points x that falls on the hyperplanes satisfy the following equation

$W.X - b = 0$…………………… eqn (1)

Where "." denotes the dot product and W is normal to the hyperplane. The parameter $\frac{b}{|W|}$ determines the offset of the hyperplane from the origin along the normal vector W. For linearly separating data, we can select two hyperplanes perfectly classifying the data, and then try to maximize their distance. The region bounded by them is called "margin". These hyperplanes can be described by the equations shown below

$\mathrm{W.X - b = 1}$……H1…….. eqn (2)

and

$\mathrm{W.X - b = -1}$……H2……… eqn (3)

By using geometry, the distance between these two representing hyperplanes is $\frac{2}{|W|}$, the margin of $|W|$ is minimized to ensure the falling of data point into the margin.

$\mathrm{W.X_i - b \geq 1}$ for $\mathrm{X_i}$ for the first class ………. eqn (4)

or

$\mathrm{W.X_i - b \leq 1}$ or $\mathrm{X_i}$ for the second class ……… eqn (5)

This can be represented in the single form of inequalities:

$\mathrm{Y_i(W.X_i - b) \geq 1}$, for all $1 \leq i \leq n$…………….. eqn (6)

To optimize this equation we need to find the min |W| subjected to

$\mathrm{Y_i(W.X_i - b) \geq 1}$, for any $i = 1,2, \ldots, n$ ……….. eqn (7)

The optimization problem presented here depends on ||W||, and is difficult to solve. It is possible to alter the equation by substituting |W| with $\frac{|W|^2}{2}$ without changing the solution, which brings the problem into quadratic programming optimization.

So, to find min$\frac{|W|^2}{2}$,

Subjected to

$\mathrm{y_i(W.X_i - b) \geq 1}$, for any $i = 1,2 \ldots, n$ …….. eqn (8)

By introducing Lagrange multipliers α, the previous constrained problem from eqn (8) can be expressed as

$L=\frac{|W|^2}{2} - \sum_{i=1}^{n} \alpha_i Y_i (X_i \,.\, W + b \sum_{i=1}^{n} \alpha_i$ ……………... eqn (9)

We need to minimize eqn (9) with respect to W, b and maximize with respect to α_i, for all constraints $\alpha_i \geq 0$. It is done by taking first derivative of L with respect to W and b and setting derivative to zero.

$\frac{\partial L}{\partial W} = 0 \Longrightarrow W = \sum_{i=1}^{n} \alpha_i Y_i X_i$ …………………. eqn (10)

$\frac{\partial L}{\partial b} = 0 \Longrightarrow \sum_{i=1}^{n} \alpha_i Y_i$ ………………………... eqn (11)

Substituting eqn (10) and eqn (11) into eqn (9) gives the solution of dual quadratic problem,

$L=\sum_{i=1}^{n} \alpha_i - 1/2 \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \, \alpha_j Y_i Y_j X_i X_j$ ……….. eqn (12)

Subjected to

$$\sum_{i=1}^{n} \alpha_i Y_i = 0, \qquad \text{for } \alpha_i \; \geq 0, \text{for every i}$$

The goal is to maximize L. each α_i with α_1>=0 indicates the corresponding X_i is a support vector. Then the decision function can be expressed as:

$f(x) = \sum_{i=1}^{n} \alpha_i X X_i Y_j + b$ ……………………... eqn (13)

2.3.10.2 Linearly non-separable data

In case of linearly inseparable data, the strategy of constructing the optimal plane that bisects the dataset is not applicable. An example of this type is shown in Figure 2-9. However, if this data producing error is removed then the same strategy will work as before which is called as soft margin method. Soft margin relaxes the margin constraints by pushing some data points into another side of hyperplane that splits the data points as cleanly as possible. To measure the degree of misclassification of data X_i, it presents non-negative slack variables ξ_i as shown in Figure 2-10.

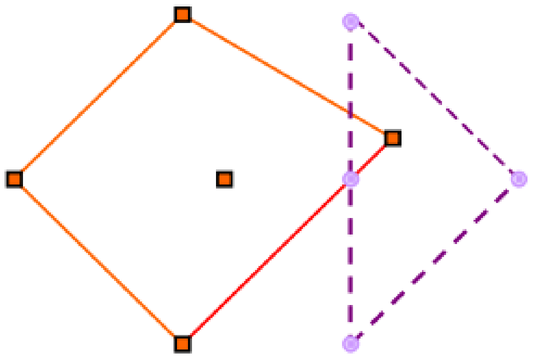


Figure 2-9: Linearly inseparable data

Figure shows a case with linearly inseparable data. Modification in the linear discriminant algorithm is needed to solve this type of case, taken from [71]

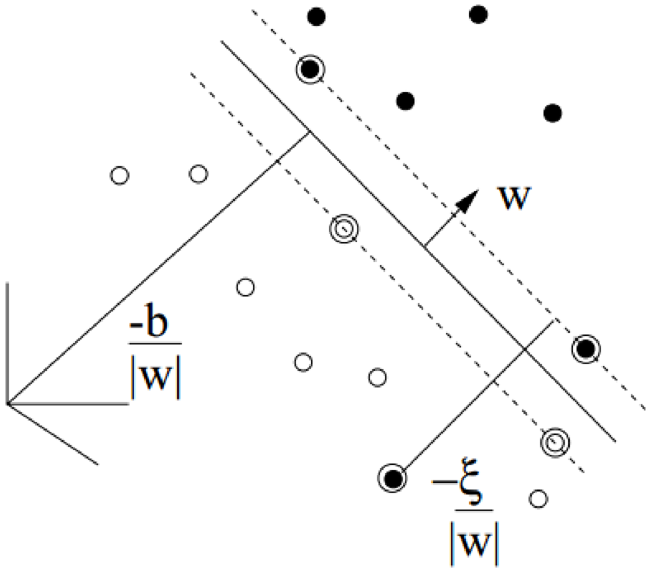


Figure 2-10: Mathematical representation linearly inseperable data

Figure shows the modification done to fit linearly inseparable data using the same algorithm as linear discriminants (soft margin method), taken from [73]

This is given by:

$$X_i.W + b \geq +1 - \xi_i \ \text{ for } Y_i = +1 \ \ldots\ldots\ldots\ldots \text{ eqn (14)}$$

$$X_i.W + b \leq -1 + \xi_i \ \text{ for } Y_i = -1 \ldots\ldots\ \ldots\ldots \text{eqn (15)}$$

$$\xi_i \ \geq 0 \ \text{ for every i } \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{ eqn (16)}$$

It can be written as

$Y_i(W.X_i - b) \geq 1 - \xi_i$, where $1 \leq i \leq n$….. eqn (17)

There exist a tradeoff between the large margin and error penalty. For the linear error penalty function the problem becomes

$\min_{w,b,\xi} L = \frac{||W||^2}{2} + C \sum_{i=1}^{n} \xi_i$………… eqn (18)

Subjected to for any $i = 1,2 \ldots., n$

$$Y_i(W.X_i - b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

This constraint in (2) along with the objective of minimizing $||W||$ can be solved using Lagrange multipliers as done above. One has then to solve the following problem:

$\min_{w,b,\xi} \max_{\alpha,\beta} \left\{ \frac{||W||^2}{2} + C \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i \, [Y_i(X_i.W - b) - 1 + \xi_i] - \sum_{i=1}^{n} \beta_i \xi_i \right\}$….eqn(19)

$$\text{for } \alpha_i \beta_i \geq 0$$

Where “C” is the regularization parameter and responsible for tradeoff between training error and complexity term. For the linearly separable case this optimization problem can be converted into dual problem as done above.

Maximize (in α_i)

$\tilde{L}(\alpha) \sum_{i=1}^{n} \alpha_i - 1/2 \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \, Y_i Y_j k(X_i, X_j)$…………… eqn (20)

Subjected to for any $i = 1,2, \ldots., n$

$$0 \leq \alpha \leq C$$

And

$$\sum_{i=1}^{n} \alpha_i Y_i = 0$$

The constant C is only the difference in this equation than compared to previous equation that adds additional constraint on the Lagrange multipliers.

2.3.10.3 Nonlinear functions via kernels

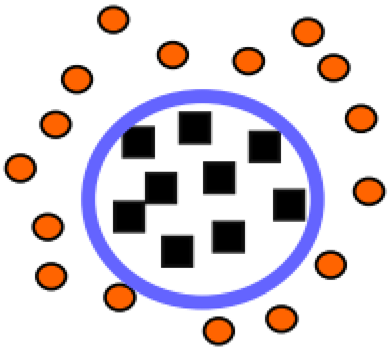


Figure 2-11: Quadratic discriminant

Figure shows a case of quadratic discriminant that cannot be classified by linear classification algorithm, taken from [71]

In case of quadratic discriminant, the decision function is not the linear function of data and thus above mentioned strategy will not fit. Therefore, conversion from linear classification algorithm to nonlinear classification algorithm done by adding additional attributes to an original data. For this purpose Boser et al., in [74] , used kernel substitution also known as kernel trick. No change in algorithm has to be made by changing kernels which will turn into general nonlinear algorithm. This nonlinear classifier can be used to train nonlinear functions like polynomial, sigmoidal neural network etc.

By mapping each data points from eqn 20 into high dimension space through transformation ø such that

$X \rightarrow \phi(x)$, then the dot product becomes

$$K(X_i, Y_j) \rightarrow \phi(X_i)\phi(Y_j) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{ eqn (21)}$$

Here, the function $K(X, Y)$is called kernel function.

Then,

$$L = \sum_{i=1}^{n} \alpha_i - 1/2 \sum_{i=1}^{n} \sum_{j=1}^{n} a_{i_i} \; \alpha_j Y_i Y_j k(X_i, X_j) \ldots\ldots\ldots\ldots \text{ eqn (22)}$$

Subjected to

$$\sum_{i=1}^{n} \alpha_i Y_i = 0$$

$$0 \leq \alpha \leq C$$

Then, the decision function is given by $f(x) = \sum_{i=1}^{n} \alpha_i Y_i k(X_i, X_j) + b$….eqn (23)

The performance of SVM is dependent to the selection of suitable kernel function. Some of the kernel functions that are available in Weka are listed in the Table 2-4.

Table 2-4: Kernel for classification using SVM and its equation

There kernels were used to in this project to classify the real and false binding sites using SVM.

S.n.	Kernel type	Relation
1	polynomial kernel	K(x, y) = <x, y>^p or K(x, y) = (<x, y>+1)^p
2	Radial basis function (RBF)	K(x, y) = e^-(gamma * <x-y, x-y>^2)
3	PUK kernel	The Pearson VII function-based universal kernel

2.3.11 Other classification approaches

2.3.11.1 Random Forest

Random forest is an ensemble of decision tree developed by Leo Breiman [75] . This algorithm for classification uses bootstrapping for building a tree and grows an unpruned tree. In other standard tree bases classification approaches, node is split based on the best split among all variables. However, to split each node in random forest, randomly chosen best predictor at that node is considered [76] . RF uses ensemble and bootstrapping technique to prevent over fitting.

Features of random forest:

- Feasible to use even though the number of variable is larger than number of observations
- It can be used to classify both two class or multi-class problems

- It so not suffer from over fitting
- It has best accuracy among current algorithms
- Can be used in large dataset

Random forest has been used in many classification purposes in genome study. In [77] , RF was used for classification of microarray data. They have shown that RF has good performance and accuracy as other machine learning methods like KNN and SVM. The extension of RF has also been used in microarray data classification [78] [79] .

2.3.11.2 Naïve Bayes

Naïve Bayes classifier is based on Bayes' theorem with Naïve independence assumptions between the features. Naïve Bayes classifier is based on a statistical as well as supervised learning. It has been used in classification of text, spam etc. [80] . Naïve Byes classifier can be used with small amount of training data. It has been used in bioinformatics in different classification task. In [81] , Gail et al. implements Naïve Bayes classifier (NBC) to classify taxonomic match of metagenomic reads. Similarly, in [82], extension of Naïve has been used for the selection and classification of SNP data.

Chapter 3

3 Approach and Methodology

3.1 High Level Design

Figure 3-1 shows a work flow diagram for this project. A ChIP-seq region of true binding site for USF1 was provided in bed file format in together with the motif describing those binding sites. This file was used to make positive and negative dataset, which was finally used to make a test and training set. To make positive dataset, the bed region described by true motif was expanded on both sides. A fasta format was extracted for the expanded region. This obtained fasta file was used to calculate the various physical properties like stacking energy, DNA denaturation, BDNA twist and protein DNA twist . The statistical measures like mean, standard deviation, kurtosis and skewness were calculated from each physical property. In addition, genomic features like GC count, GC skew, distance from TSS to binding region and distance from CpG Island to binding region were computed.

Similarly, to make negative dataset, a region 500 base pair far from the true binding site was selected. This region was used as input to FIMO together with motif describing the binding region. The output of FIMO returns the regions that are potential binding site but are actually false positive value. These coordinates of false positive motif were then trimmed and aligned. Then finally the region was extended on both sides making the distribution same as that of positive dataset. Fasta for these regions were calculated in the similar way as before. Then, this fasta file was used to calculate the physical properties. Finally, the statistical measure of these properties was calculated. Last but not the least, the genomic features like TSS distance, CpG island , GC skew and GC count were also calculated, which complete the negative dataset. After solving the imbalance problem of positive and negative dataset, a single file was made mixing positive and negative dataset. From the file, 90% of instances were used as training set and remaining 10 % as testing set. The entire flow diagram is shown in figure 3-1. Note: The duplication of data between test and training set was strictly avoided.

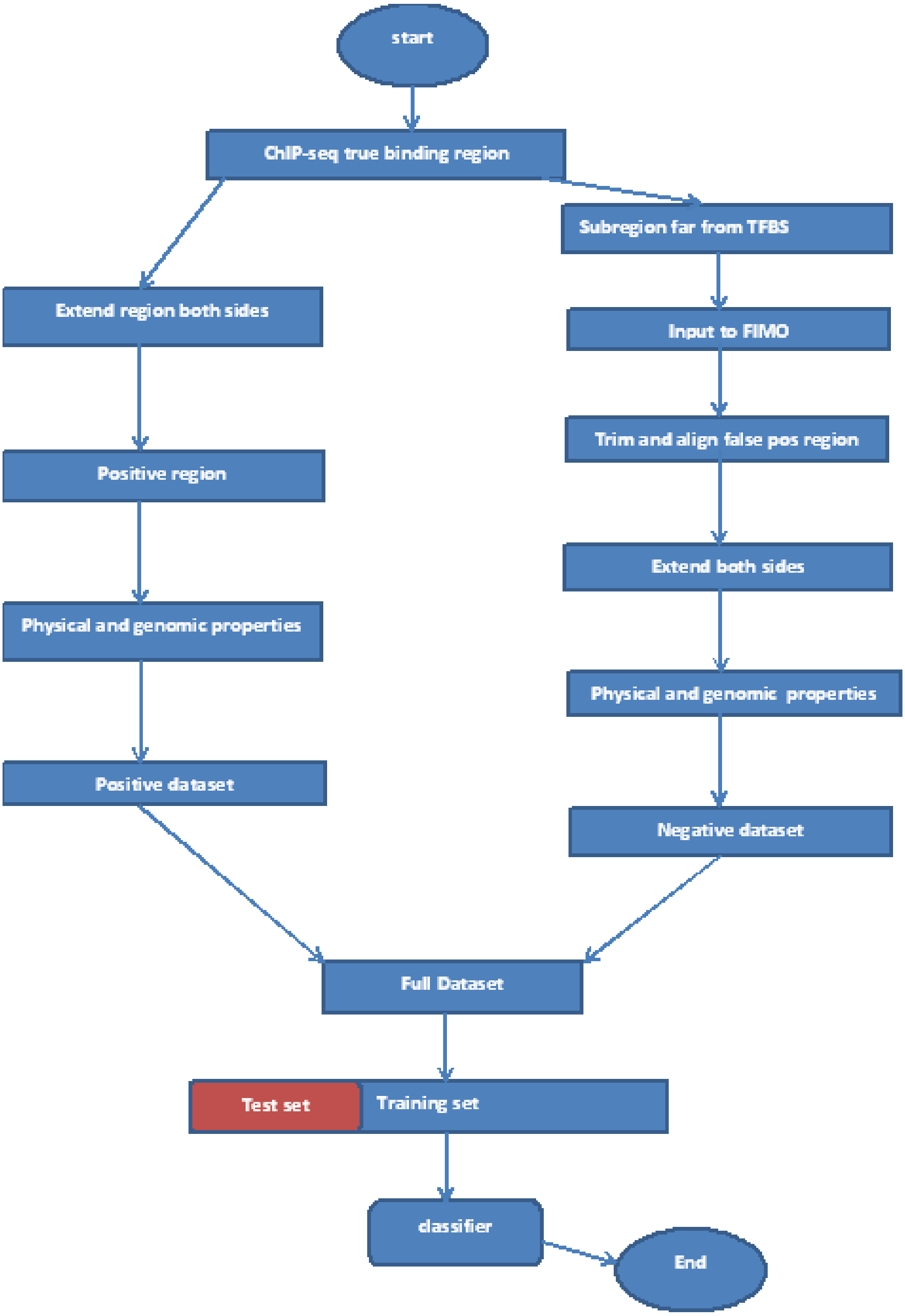


Figure 3-1: Flow diagram of RAFT

3.2 Algorithms

This section contains the algorithms and pseudo-code that were developed to compute various properties using Java program.

3.2.1.1 GC count

This is the algorithm written in Java to count the GC content in DNA sequence. The output of this program will give the percentage of GC. It will input the single file in fasta format, compute GC content line by line from file and then generates the output.

```
Step 1: Start
Step 2: Get input (x) from file in fasta format one line at a time
Step 3: Declare variables x, y, j, count, result;
Step 4: Initialize variables count←0, j←0
Step 5: Calculate count of G and C
            y←x.toUpperCase();
            Repeats the step until j=x.length()
                If      y.charAt(j)='C'| y.charAt(j)='G'
                        count ←count+1;
Step 6: Limit the decimal up to 2 digits
            result← (count*100)/x.length();
            result←Math.round(result*100.0)/100.0;
Step 7: Print result
Step 8: Go to step 2, until line in the file ends
Step 9: Stop
```

3.2.1.2 GC Skew

This algorithm was developed to compute the GC skew in DNA sequence. The output of this program will gives the GC skew. It will input the single file in fasta format, compute GC skew line by line from file and then generates the output.

```
Step 1: Start

Step 2: Get input (x) from file in fasta format one line at a time

Step 3: Declare variables x, y, j, countG, countC, result;

Step 4: Initialize variables countG←0, countC←0, j←0

Step 5: Compute count of G and C

                y←x.toUpperCase();

                Repeats the step until j=x.length()

                    If          y.charAt(j)='G'
                                    countG ←countG+1

                    Else If    y.charAt(j)='C'
                                    countC←countC+1
Step 6: Compute GC Skew

                    result← (countG-countC)/(countC+countG))

Step 7: Limit the decimal up to 2 digits

                result ← (count*100)/x.length();

                 result ←Math.round(result*100.0)/100.0;

Step 8: Print result

Step 9: Go to step 2, until line in the file ends

Step 10: Stop
```

3.2.1.3 Statistical parameters

This algorithm was developed to calculate the property BDNA Twist. Similarly, values of other properties like stacking energy, protein DNA twist, and protein induced deformation were also computed by just changing the value of dinucleotide shown in Table 2-2 for respective properties. I have used a library developed by Tim O’Brien available in [83] , to compute the statistical measures like Mean, Skewness, Standard deviation and Kurtosis.

```
Step 1: Start
Step 2: Get input (x) from file in fasta format one line at a time

Step 3: Declare variables  j, valMean, valSD, valSkew, valKurt, value, result, str, values[]

Step 4: Initialize variables str←null, valMean←0, valSD←0, valSkew←0, valKurt←0, value←0,

        double[]←  new double[x.length()-1]

Step 5: Compute the value of dinucleotide properties

                Step 5.1: Parse the DNA sequence into dinucleotide with sliding window approach
                Repeats the step until j=x.length()
                        str←x.substring(j,j+2);
                        str←str.toUpperCase();

                Step 5.2: Assign the value of dinucleotide segment

                        If str.equals("AA")

                                value←35.5

                        Else If str.equals("AC")
                                value←33.1;
                                .
                                .        Calculate for all dinucleotide combination
                                .
                        Else If str.equals("TT")
                                value←35.5;

                Step 5.3: Hold the value in array

                        If value≠0

                                values[j] ←value

Step 6: Compute Statistical measures from the dinucleotide values in array

                Mean mean=new Mean()
                valMean← mean.evaluate(values)

                StandardDeviation stdDev = new StandardDeviation()
                valSD← stdDev.evaluate(values)

                Skewness skewness = new Skewness()
                valSkew← skewness.evaluate(values)

                Kurtosis kurtosis = new Kurtosis()

                valKurt← kurtosis.evaluate(values)


Step 7: Limit the decimal up to 3 digits for each returned value
```

```
                resultMean ←Math.round(valMean*1000.0)/1000.0
                resultSD ←Math.round(valSD*1000.0)/1000.0
                resultSkew ←Math.round(valSkew*1000.0)/1000.0
                resultKurt ←Math.round(valKurt*1000.0)/1000.0
Step 8: Print result
Step 9: Go to step 2, until line in the file ends
Step 10: Stop
```

3.2.1.4 Correlation calculations

This algorithm was developed to count the nucleotides at specific position and total background distribution of mono nucleotides. These obtained values were used to calculate Pearson correlation.

```
Step 1: Start
Step 2: Input file line by line and store in array "aContentArr"
Step 3: Declare arrayList mononuc, prevVal
Step 4: Initialize variables prevVal←0
Step 5: Add mononucleotides as an element to arrayList
        mononuc.addAll(Arrays.asList("A","T","G","C"))
Step 6: Declare hashmap and initialize to zero
        HashMap<String, Integer> hm = new HashMap<String, Integer>();
                hm.clear()
Step 7: Count the number of mononucleotides present in each column
        Repeats the step until i equals to length of sequence in array
                        Repeats the step until depth equals to depth of array
                        Step 7.1: Get single nucleotide and convert to uppercase
```

```
                    s←aContentArr.get(depth).substring(i, i+1);
                    s←s.toUpperCase();
                Step 7.2: Check if substring present in array and update count
                If mononuc.contains(s)
                    If hm.containsKey(s)
                    prevVal←hm.get(s).intValue()
                    hm.remove(s)
                hm.put(s,prevVal+1)
                end depth loop
                Step 7.3: Print key and value from hashMap
                Step 7.4: clear hashMap
            End i loop
Step 8: Stop
```

In this algorithm, it is important to note that hashmap value is initialized to 0. This will force the program to start the count from zero for each sequence in an array.

In similar way, background distribution of mono nucleotide can be computed. However, the difference in algorithm is that the hashmap value is not cleared (reset) in background distribution calculation. This will update the count with previous count and yields the total distribution of mononucleotides present in entire sequences. These values of mononucleotides at specific position and background distribution were exported to excel and the correlation was calculated. Based upon the correlation the plots were made that are shown in next chapter.

In addition, di-nucleotides and tri-nucleotides were also computed with simple modification in above algorithm. The elements in array for di-nucleotides and tri-nucleotides were used in place of mono nucleotides. The possible di-nucleotides and tri-nucleotides combinations are shown in Appendix . The values obtained were again exported to excel to calculate the correlation and finally the plot was made for di-nucleotides and tri-nucleotides are shown in next chapter.

3.2.1.5 Cruciform structure

This algorithm was developed to check the transcription factor forming the cruciform structure. In this algorithm, the complement of sequence is computed from the sequence. Sequence and its reverse complement is assigning in row and column of the matrix. Similar with the dot matrix approach, value 1 is added to the value of cell if the base pair in i and j are same. This is loop over all sequences present in file.

```
Step 1: Start
Step 2: Input file (x) a line at a time
Step 3: Declare x, arrayList, an, cn, tn, gn, complement, comp, revComp, i, j
Step 4: Initialize variables s←null, comp←null, revComp←null, i←0, j←0
Step 5: Add elements to array
        an.add("A")
        cn.add("C")
        tn.add("T")
        gn.add("G")
Step 6:  Find the reverse complement of sequence
        Step 6.1: change sequence to uppercase
                x←x.toUpperCase()
        Step 6.2: Find the complement of sequence
                Repeat i until length of sequence
                s←x.substring(i, i+1)
                        If  an.contains(s)
                        comp←replace A with T
```

```
         Else If tn.contains(s)
         comp←replace T with A
         Else If cn.contains(s)
         comp←replace C with G
         Else If gn.contains(s)
         comp←replace G with C
      complement.add(comp)
      end loop
   Step 6.3: Convert array to string
   comp←complement.toString
   Step 6.4: Reverse the string
   revComp←reverse(comp)
Step 7: Use the original sequence in row and its reverse complement in column and fill the cell of table, add 1 in cell if the nucleotides in i and j are same, else add 0.
   Step 7.1: Assign row with original sequence and column with its reverse complement
   Step 7.2: For each i and j position in the cell, where i is the length of original sequence and j is the length of reverse complement sequence
      If nucleotide at i equals nucleotide at j
      table[i][j] = 1+table[i][j];
      Else
      table[i][j] = 0+table[i][j];
Step 8: Go to 2 until the end of line in file
Step 9: Print the table
Step 10: Stop
```

3.3 Methodology

3.3.1 Description of Data Sets

3.3.1.1 Data sources

A file "K562USF1wgEncodeAwgTfbsHaibK562Usf1V0416101UniPk.bed" containing binding regions for the transcription factor USF1 determined using ChIP-seq was provided. This file was originally downloaded from UCSC table browser (http://genome.ucsc.edu). Another file, K562USF1wgEncodeAwgTfbsHaibK562Usf1V0416101UniPk.mat was a PWM in MEME format describing USF1 binding sites. It was determined using tool MEME from the ChIP-seq regions indicated by the binding regions for the transcription factor USF1. Finally, K562USF1wgEncodeAwgTfbsHaibK562Usf1V0416101UniPk_tfbs.bed was a list of likely TF binding sites in the binding regions. It was determined by using FIMO (Find Individual Motif Occurrences) to scan the ChIP-seq regions with the matrix, and keeping the most significant hits. Not all regions will have TF binding sites in this list. The files were for the cell type K562. The tracks were made from version hg19 of the genome.

The rest of information like CpG Island and TSS distance were downloaded from UCSC table browser in bed file format.

3.3.1.2 Features selection

For the property-based classification, different genomic and physical properties were used. The statistical measure for physical properties was calculated. These properties were used to label the positive and negative cases finally forming test and training set. Genomic and structural properties were already described in Chapter 2.

3.3.2 Positive data

ChIP-seq region for the true positive region was provided in bed file format along with the PWM description of motif. These sites defined by bed file fall in the ChIP-sec region and believed to be a true positive binding site. This bed file contains the coordinate of true binding site defined by motif and the length was exactly equal to the length described by motif that was 11 nucleotides. These bed regions were then extended to on both sides by 30 nucleotides using bed command, thus, giving constant 71 residues for each sequences (30 nts left +11 nts binding site + 30 nts right). Finally, fasta was extracted from the extended bed file. With this approach the binding site are perfectly aligned starting from 31th positions to 41th position (11nts long) of each fasta line. The properties obtained from this data were used as main source for the positive dataset. With the similar steps another positive dataset with sequence length of 161 nts was extracted. It was made by extending the true binding site by 75 nts both sides (75+11+75). This region was used to predict the appropriate window size.

3.3.3 Negative data

Again, true positive dataset was then used to generate a new negative dataset. It was not an easy task to generate a good negative data set that shouldn't be including positive binding regions. To make negative data set, a sub region of 300 nts that was 500nts far away (right side) from the coordinates defined by positive motif was chosen. There were two reasons for choosing sub-region of 300 nts in this project. With small region it was difficult to get enough false positive sequences defined by motif in this sub-region. If the larger sub-region was taken then there was possibility of getting large number of false positive sequences, however, there is limitation of maximum 1 million sequence characters as input. The sub-region used in this project was acceptable with FIMO's limitation of maximum number of characters as input and was able to predict enough number of motifs around false positive region. The fasta format was extracted using the bed command for this new region. Then, the fasta region was used as input for the FIMO along with the description of motif. Finally, FIMO returns the information with list of coordinates and the binding motif. It is important to note that, FIMO use default threshold of e^{-4} therefore, the number of output may not me equal to number of input coordinates. The hits returned by FIMO are generally less in false positive region than number of positive motifs.

The output was FIMO was then trimmed to obtain only 11nts false positive binding region. This false positive binding region was then extended by 30 nts both the direction making total of 71nts long sequence. The fasta for this region was obtained using the bedtool. This fasta region was 71 nts long in which false positive binding region starts from 31th position to 41th position and of course, same length as positive binding site. These false positive regions did not contain the region described by ChIP-seq. This was confirmed by scanning the motif using FIMO which returned the list of region that did not fall in ChIP-seq region. Finally, the properties of these regions were used to train the model as negative training set.

I also tried to make another false region with respect to false positive region in order to check the variation in between these two regions. This new false positive region generated from false positive is termed as false_false positive in this report. With similar approach this false_false positive region was made. A sub-region of 400 nts was selected 200 nts away (right side) from the motif found in false positive region. Fasta from this sub region was used as input to FIMO with the motif defining binding site. The FIMO returns 82 hits that were 7.2 % of false positive set. This region was extracted aligned and extended on both sides by 30 nts forming false_false positive region. This false_false positive region has not been used in classification purpose but it has been used to compare with false positive dataset using sequence logo.

3.3.4 Imbalance problem

The large variation in the number of positive and negative cases gives the bias prediction. These will give over fitting problem and mislead the prediction. To avoid this problem a mechanism to balance the dataset was done. Two approaches were suitable to balance the dataset.

First, SMOTE function of WEKA is inbuilt function in weka that provides the option to automatically balance the minor member present in training set using supervise learning. Second approach is to randomly sample the large data set (positive dataset in this case) and using the subset of balance dataset. In this thesis, later approach was used. Positive case was large in number than negative case. Therefore, random sampling of positive cases was done to make a subset of positive cases. This newly formed dataset was balance set that was used to make test and training set.

The redundancy and inconstancy was not observed in positive data set. However, few redundancy and inconstancy was observed in negative dataset. Some of the sequences in the file were less than 71nts long was observed. This was due to the fact that when selecting the region 500 nts away from the true binding site, some of the regions came close to the end of DNA sequence for that particular strand. There were 38 such inconsistent sequences that constitute 3.5% of total negative set. These redundancy and inconstant region were discarded from the negative dataset for further analysis.

3.3.5 Pre-processing

The genomic and structural features of both positive and negative dataset were computed. These computed attributes of positive and negative data were then merged into a single file. The row was randomized to shuffle the positive and negative cases. The file was then used to make a test and training dataset. To make a test dataset 10% of total cases were used. The remaining 90% of cases were used as training set. Finally, the file was converted in to ARFF file format by a converter to feed into classifier. The distribution of properties of total cases i.e. before separating test and training set are shown in Figure 3-2. The positive instances are shown in red and negative instances are shown in blue. Figure shows the aggregation of positive and negative cases around the mean value forming a peak.

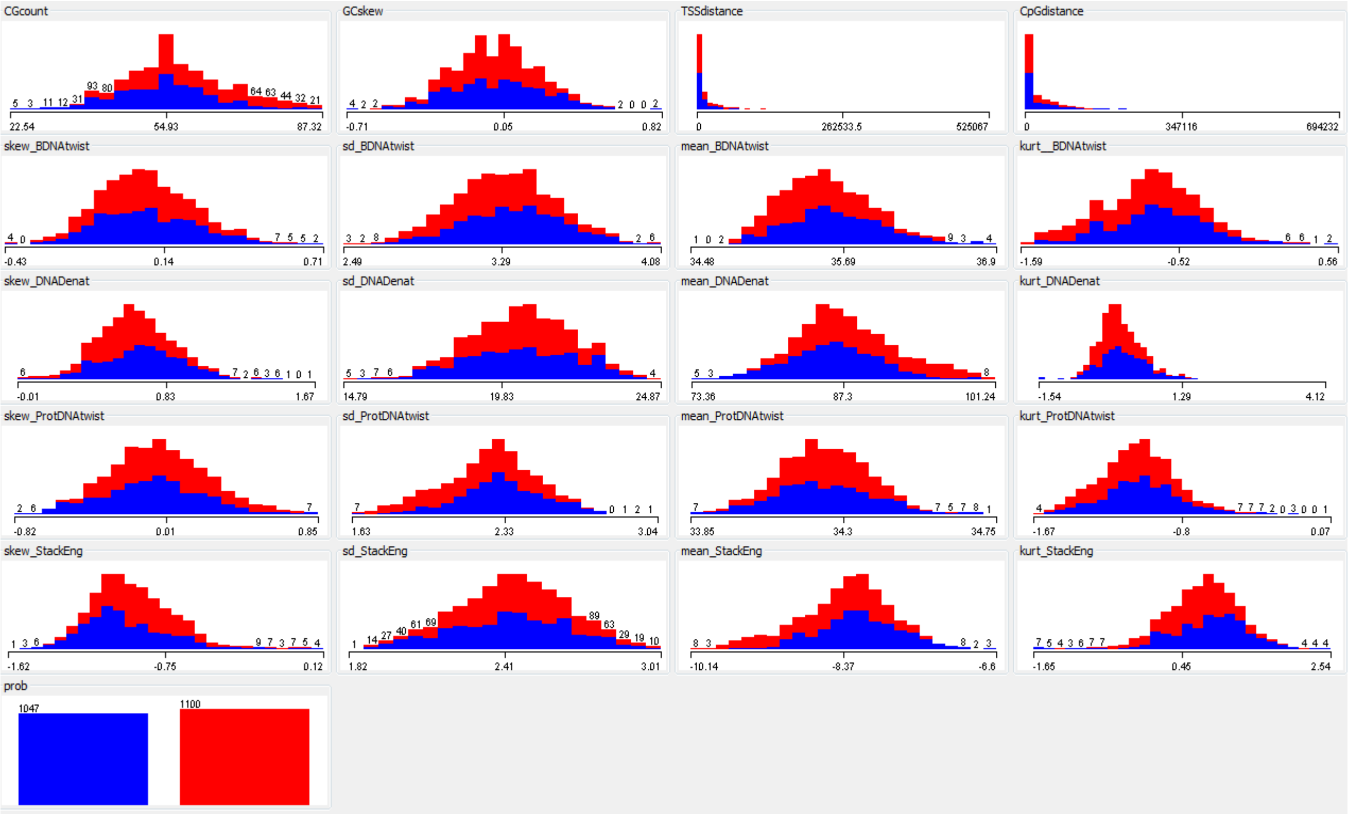


Figure 3-2: Relative distributions of properties

Figure shows a relative distribution between positive and negative instances (red positive instances and blue negative instances). It demonstrates that both instances forming peak around the mean value.

3.3.5.1 Training set

It was assumed that 90% of total data was sufficient to use as training set. Therefore, 1932 instances were used as training set which was exactly 90% of total file (2147 instances). These, contains 962 positive instances and 970 negative instances.

3.3.5.2 Test set

To make test set I used 10% of dataset that were 215 instances. These instances comprised of 118 positive and 97negative instances. These training and test set were then ready to feed the classifier.

3.3.6 Quality assurance of dataset

To ensure the quality of data correlation tests was done. Firstly, I computed the correlation between background distribution and actual distribution at that point. It was done to test whether the region around the TFBS was reasonable or not. Since, if the window is too small then it is possible to loose important information. In similar way, if the window size is too large then any real signal may be lost in additional noise. Following steps were followed to compute this correlation assuming mononucleotides. This was done in full positive dataset before making the subset.

- Loop over full positive dataset and count the number of A, C, T and G at each position of the region.
- Loop again and count the number of bases independent of position that gives the background distribution.
- Now for each position of the profile, estimate the correlation between the background distribution and actual distribution at that position.
- Make a correlation plot which indicates the different parts of these regions deviates from random background distribution. This plot gives idea about the signals present around the TFBS. During this computation the region including motif gives biased composition.

The correlation was also computed for dinucleotide and tri-nucleotides since they have more data points for correlation than in mononucleotides.

3.3.7 Classifier and its performance

The support vector machine (SVM) was trained using SMO on the training data. Different kernels were used to find the optimum accuracy of classifier. Simply taking an account of accuracy for the performance of classifier is not sufficient thus other parameters like sensitivity, specificity, F1-score and accuracy were also calculate for better measure of classification performance.

3.3.7.1 F1 -score:

It is the harmonic mean of precision and sensitivity.

$$F = 2 * Precision * Sensitivity/(Precision + Sensitivity)$$

3.3.7.2 Sensitivity (recall) or true positive rate

Sensitivity is defined as fraction of correctly predicted case.

$$Sn = TP/(TP + FN)$$

3.3.7.3 Specificity –false positive rate

$$Sp = TN/(TN + FP)$$

3.3.7.4 Accuracy:

$$A = (TP + TN)/(TP + TN + FP + FN)$$

Where naming are: TP- true positive, TN- true negative, FP – false positive and FN- false negative.

3.4 Resources and tool used

3.4.1 BED tools

The BED tools are fast and flexible tools for testing for manipulation and analysis between different large set of genomic features [84] . This tool was also described in spring project report. The most of the description is redundant since the tool and command is very useful in accomplishing this project. It is implemented in C++ and freely available online (website of bed). In RAFT, bedtool version 2.19.1 was used. The BED tools are used to perform different operations like intersects, count, merge, expand window, subtract on different file formats like BD, BAM, VCF and GF /GTF. BED format has three mandatory fields and nine additional optional fields. The first there mandatory fields are:

chrom: this is the mandatory field that is used to name chromosome. Any string can be used to name the chromosome.

chromStart: the zero bases starting position of the features in the chromosome. The first base of chrosome is number 0. The natural numbers are used to indicate the chromStart.

chromEnd: this is another mandatory field which specifies the one based ending position of the feature in the chromosome. Similar to the chromStart, numbers are used to indicate the end position of chromosome.

There are other nine additional BED files like name, score, strand, thickStart, thickEnd, itemRbg, blockCount,blockSize and blockStarts. The descriptions for these fields are found in BED tool manual.

Some useful bed commands that are used in this project are explained in brief.

3.4.1.1 IntersectBed

This tool will report the intersection or overlapping features between two files A and B.

Usage: $ intersectBed [OPTIONS] -a A.bed -b B.bed

Option	Description
-a	BED file A
-b	BED file B
-wb	Write the original entry in A for each overlap.
-wa	Write the original entry in B for each overlap. Useful for knowing what A overlaps. Restricted by -f.
-s	Enforce to overlap the feature for same strand if present

This command will report the overlapping features in standard output. To extract the output in named file “C” we may add “>C.bed” at the end of that command. The detailed options can be found in user manual of bedtools.

3.4.1.2 closestBed

It will also searches for the overlapping features between A and B, if not found closestBed will report the closest (that is, least genomic distance) between A and feature in B.

Usage: $ closestBed [OPTIONS] -a A.bed -b B.bed

Option	Description
-s	Force strandedness. That is, find the closest feature in B overlaps A on the same strand. By default, this is disabled.
-t	How ties for closest feature should be handled. This occurs when two features in B have exactly the same overlap with a feature in A. By default, all such features in B are reported. Here are the other choices controlling how ties are handled: all Report all ties (default). first Report the first tie that occurred in the B file. last Report the last tie that occurred in the B file.

3.4.1.3 subtractBed

It searches for the features in B that overlap with the features in A and if found subtractBed will remove the overlapping portion from A and the remaining features are reported.

Usage: $ subtractBed [OPTIONS] -a A.bed –b B.bed

Option	Description
-f	Minimum overlap required as a fraction of A. Default is 1E-9 (i.e. 1bp).
-s	Force strandedness. That is, find the closest feature in B overlaps A on the same strand. By default, this is disabled.

3.4.1.4 fastaFromBed

This tool extracts sequences from a FASTA from input fasta file source for the intervals defined in a input BED file.

Usage: $fastaFromBed [OPTIONS] -fi hg19.fa -bed A.bed -fo <output.fa>

Option	Description
-names	Use the “name” column in the BED file for the FASTA headers in the output FASTA file.
-tab	Report extract sequences in a tab-delimited format instead of in FASTA format.

3.4.1.5 slopBed

slopBed will increase the window size of each feature in a BED file as defined by user. The defined number must be an integer value.

Usage: $ slopBed [OPTIONS] -i <BED> -g <GENOME> [-b or (-l and -r)]

Option	Description
-b	Increase the BED entry by the same number base pairs in both directions
-l	The number of base pairs to subtract from the start coordinate
-r	The number of base pairs to add to the end coordinate
-s	Define -l and -r based on strand. For example- if used, -l 500 for a negative-stranded feature, it will add 500 bp to the end coordinate.

3.4.2 The MEME Suite

The MEME Suite is a tool for discovery and analysis of DNA binding sites and protein interaction domains within the given sequence motifs. The MEME motif discovery uses GLAM2 algorithm that allows discovery of motifs containing gaps [85] . This unified web server interface provides four different types of services for motif analysis like; motif discovery, motif -motif database searching, motif-sequence database searching and assignment of function. MEME is implemented in ANSI C and published as Simple Object Access Protocol (SOAP- it is a specification for exchanging structured information in web service).

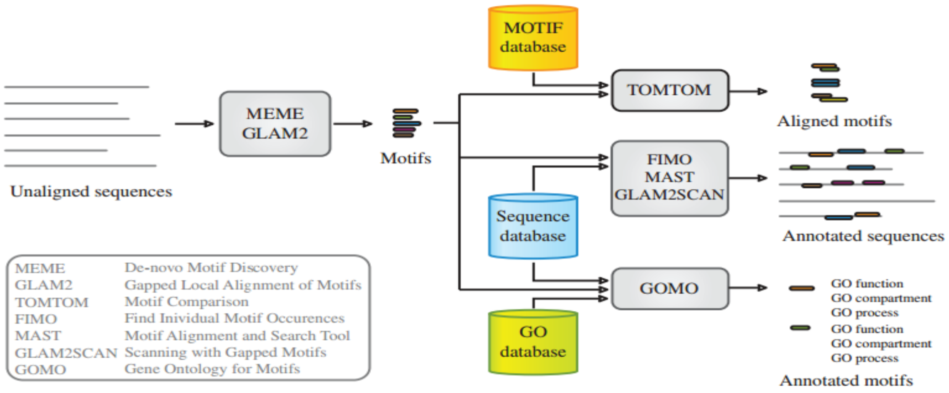


Figure 3-3: MEME suite tools

Figure 3-3 shows the function unit of MEME suite tool found in [86] . TOMTOM is a tool that is used to compare the DNA motif to a known motif in database. Its output is the score value of similarity and statistical significance of score. GOMO is used to analysis DNA motif. It searches species-specific GO annotation database. FIMO (Find Individual Motif

Occurrence) and MAST (Motif Alignment and Search Tool) are tools within MEME that are used to search sequence database. MAST is sequence oriented, is suitable to analyze fix length proteins whereas FIMO can be used to scan entire genomic database. In this project, I have work with FIMO to scan the input motif for predicting TFBSs. Therefore, I have presented FIMO in brief.

3.4.2.1 FIMO

FIMO takes as input fixed-length motifs, represented as position-specific frequency matrices [87] . The input sequence is accepted in fasta format from the interface as well. A valid Email address and support database are another required field in FIMO. The output of FIMO is a ranked list of motif occurrences with p value, q-value, start and end position of potential binding site and name of sequence. This output is represented in different ways like html, xml, CISML, GFF and plain text.

3.4.2.2 Other tools

There were other tools used to accomplish this work. Eclipse Juno Service Release 1 was used as editor for Java program. Similarly, notepad++ v5.9.6.2, Microsoft word and excel 2011 were used. In addition, web tool like Sequence2Logo was used to make a logo plot.

Chapter 4

4 Observations and analysis of outcome

4.1 Results and Discussions

This section covers the observations and outcome of the entire work. All observed results are analyzed and explained.

4.1.1 Observed dataset

First of all, there were 12334 positive sequences and based upon these sequences, 1085 (8.7%) false positive hits were obtained from FIMO in false a positive region, which was used to make negative dataset. The output of FIMO gives 38 inconsistent hits (3.5%) in negative set and those were discarded. Thus, only 1047 hits were used as negative dataset. To balance dataset between positive and negative sets, 1100 instances of positive were used, forming 2147 number of instances in full dataset. From this full dataset, 10% of instances (i.e. 215 instances) were used in test set and 90 % of instances were used in training set. These observations are summarized in Table 4-1and Table 4-2.

Table 4-1: Observation of number of positive and negative instances

	Total number of instances	**Inconsistency**	**Used instances**	**Discarded**
Positive set	12334	No	1100	11234
Negative set	1085	Yes	1047	38

Table 4-2: Observation of positive cases and negative cases in test and train dataset

Full dataset	**Test set**		**Training set**	
2147	215		1932	
	Positive cases	**Negative cases**	**Positive cases**	**Negative cases**
	118	97	962	970

4.1.2 Correlation Plot

Correlation of nucleotides at specific position with background was computed and plot was made. This plot shows the nature of distribution of nucleotides. All the plots are shown in following figures.

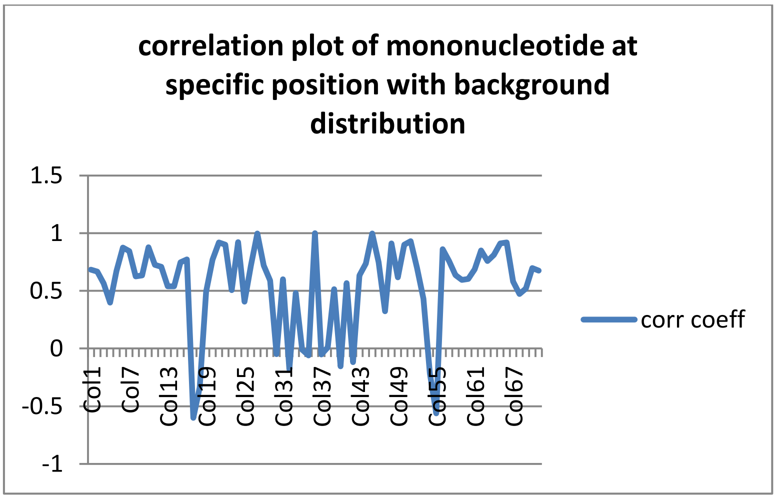


Figure 4-1: Correlation plot of mononucleotide for positive dataset

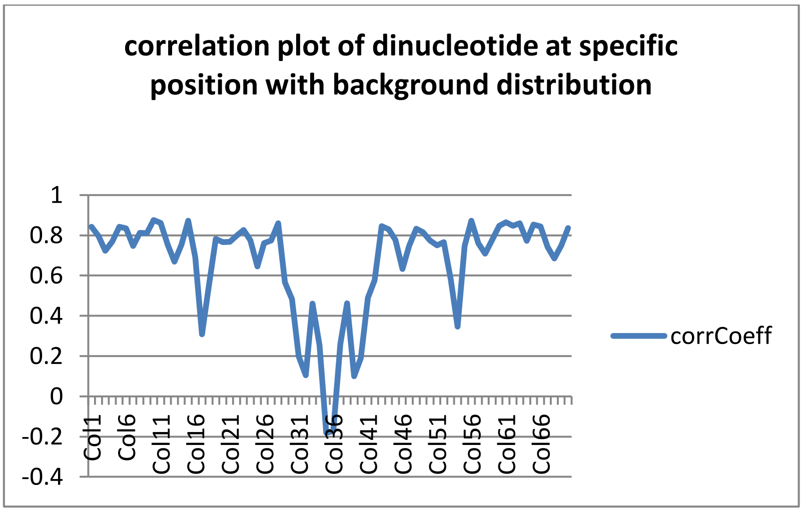


Figure 4-2: Correlation plot of dinucleotide for positive dataset

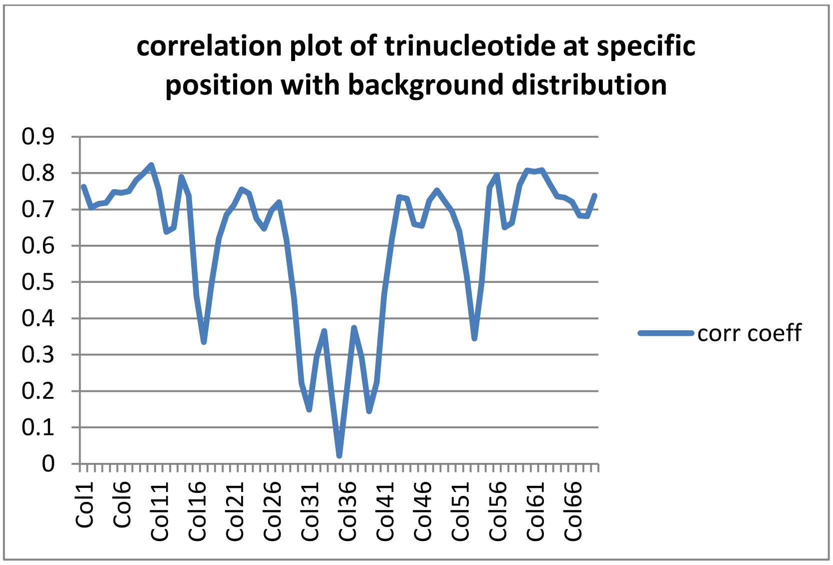


Figure 4-3: Correlation plot of tri-nucleotide for positive dataset

Figure 4-1, Figure 4-2 andFigure 4-3 show more clear signals around region 31 to 41 nts, which is the binding region. These signals are expected to exist because they form a biased composition. Interestingly, another clear signal was observed around 18 nts and 54 nts that can be seen in all three plots. This gives the idea that there might me another regulatory region regulating the binding process. The basic idea was to select the region that holds all signals around TFBS. To make sure, the region with larger extended region i.e. 100 nts both sides thus giving 211 nts long sequence were generated. The correlation was computed and the plot was made which is shown in Figure 4-4Figure 4-5Figure 4-6.

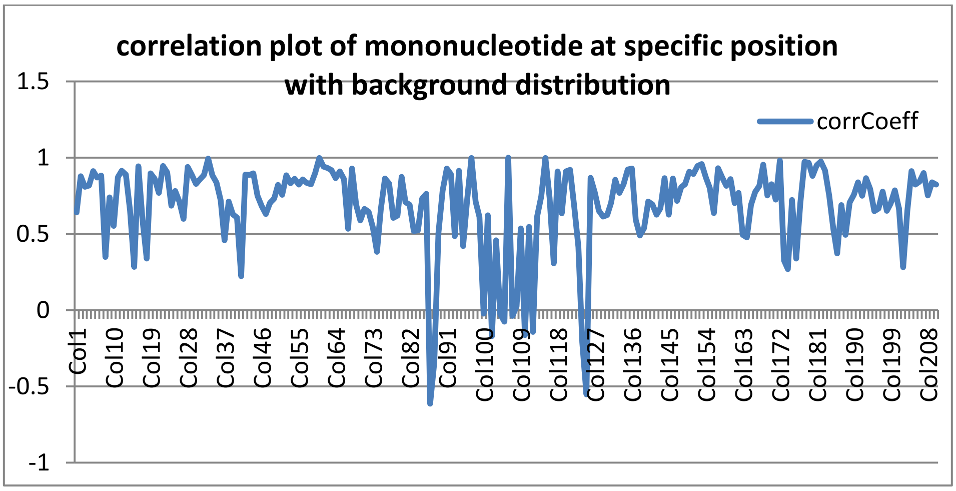


Figure 4-4: Correlation plot of mononucleotide for wider region of positive dataset

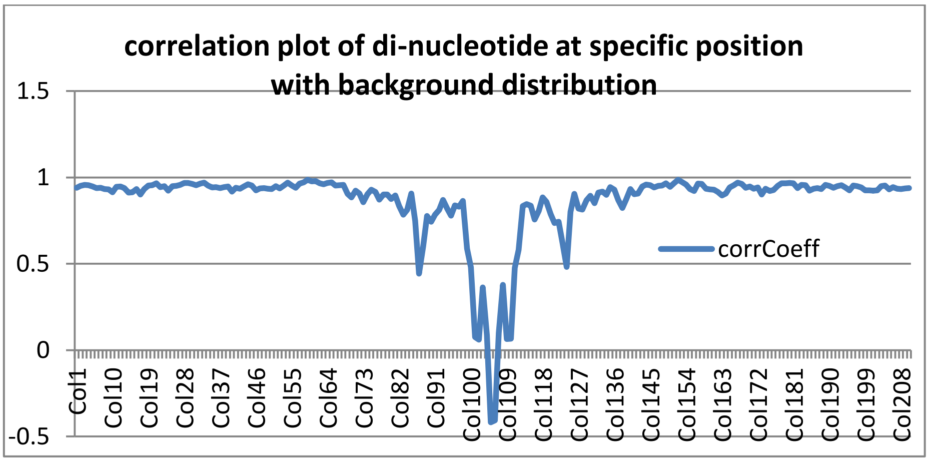


Figure 4-5: Correlation plot of di-nucleotide for wider region of positive dataset.

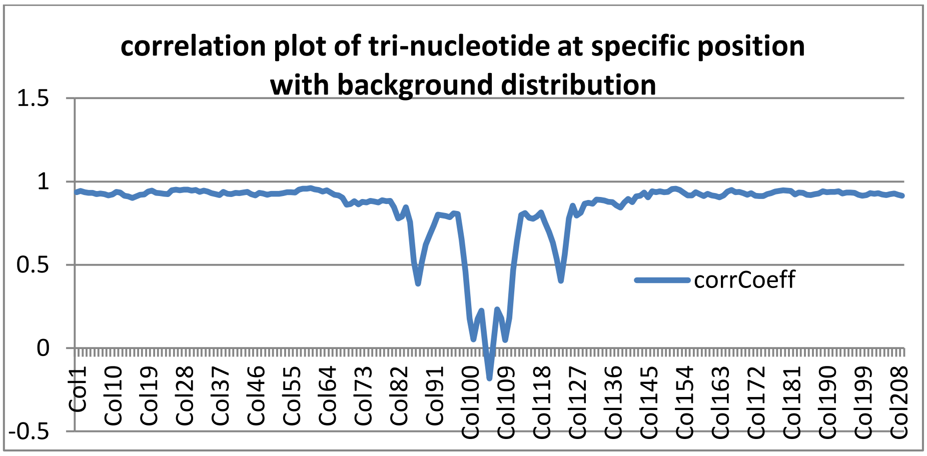


Figure 4-6: Correlation plot of tri-nucleotide for wider region of positive dataset

Comparing the plots made with wider region with 211 nts and narrow region with 71 nts. It was seen that larger region has similar correlation as narrow region with additional flat correlation in wider extended region. All the necessary signals were also conserved in narrow region towards the center of plot. Therefore, the region of 71 nts was used in this project for preparation of data and classification.

Similarly, the distribution of signals in reverse complement of sequence was also computed. The sequence was first reversed and then its complement was computed using java program. The correlation was calculated in similar fashion as mentioned earlier. The plot was made for mononucleotides, di-nucleotides and tri-nucleotides. These plots are shown in Figure 4-7Figure 4-8 Figure 4-9

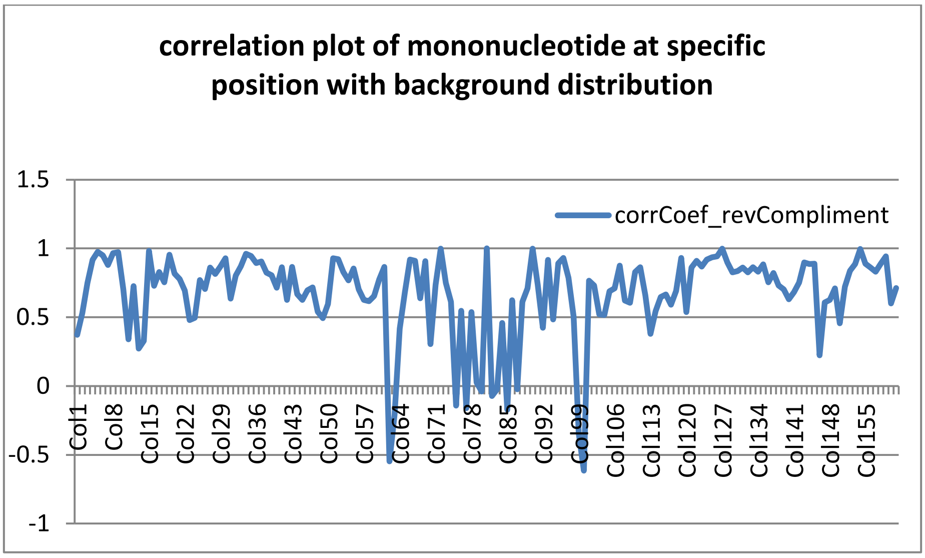


Figure 4-7: Correlation plot of mononucleotide for reverse complement of positive dataset

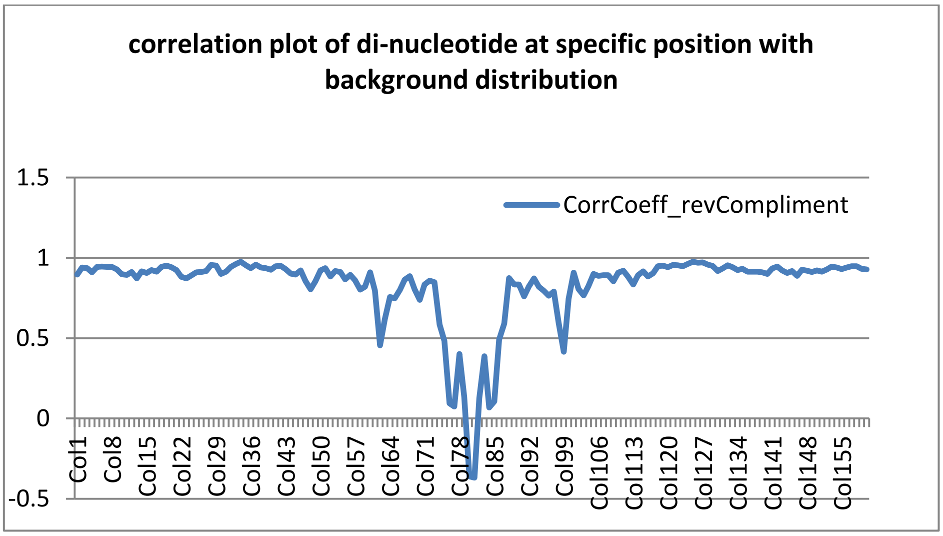


Figure 4-8: Correlation plot of di-nucleotide for reverse complement of positive dataset

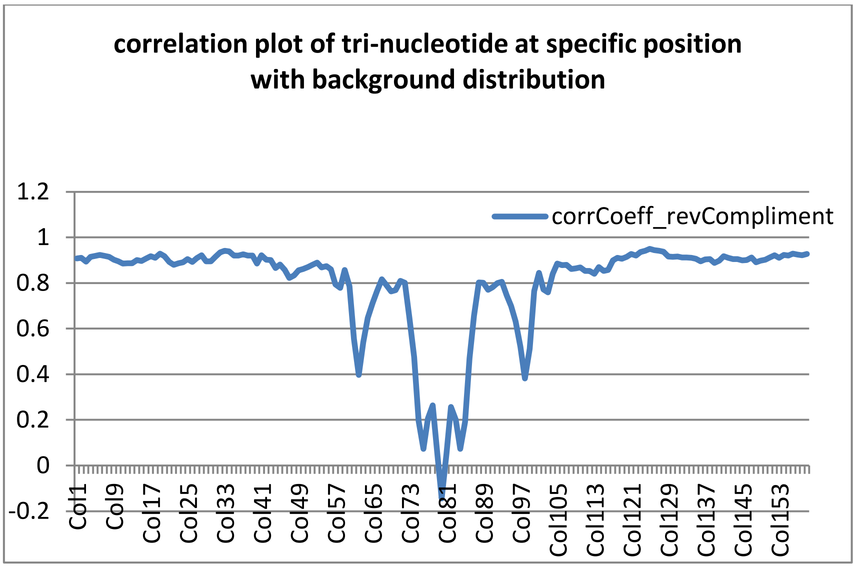


Figure 4-9: Correlation plot of tri-nucleotide for reverse complement of positive dataset

The plots of reverse complement showed the similarity with original sequence. The sharp change in correlation value can be observed in the middle of plot. This middle region is the binding site of sequence. Besides, these regions distinct occurrence of signals in both the sides of binding sites was observed. These signals were also present in narrow and wide version of plots.

Furthermore, the plot was made for reverse complement of positive strand and reverse complement of negative strand. The sequences from positive regions were first discriminated into different sets of negative and positive strand. Then, each individual set was used to make the reverse complement. The plots are presented in Figure 4-10, Figure 4-11Figure 4-12.

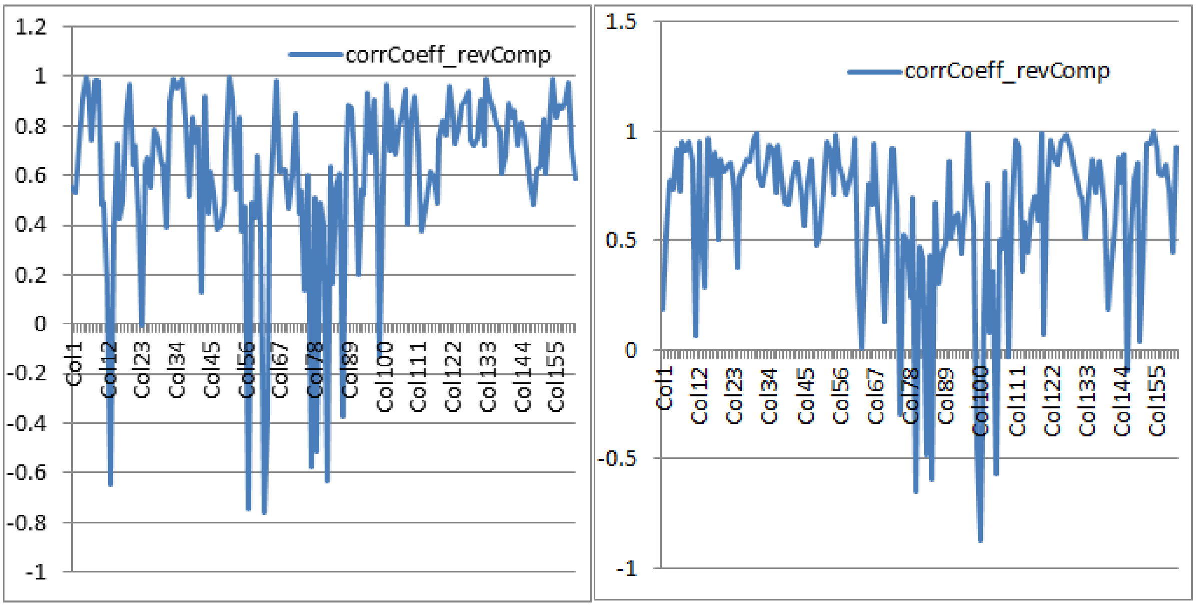


Figure 4-10: Correlation plot of mononucleotide considering strandness of reverse complement

Figure shows a correlation plot of mononucleotide at specific position with background distribution. (a) Reverse complement of negative strand (b) reverse complement of positive strand

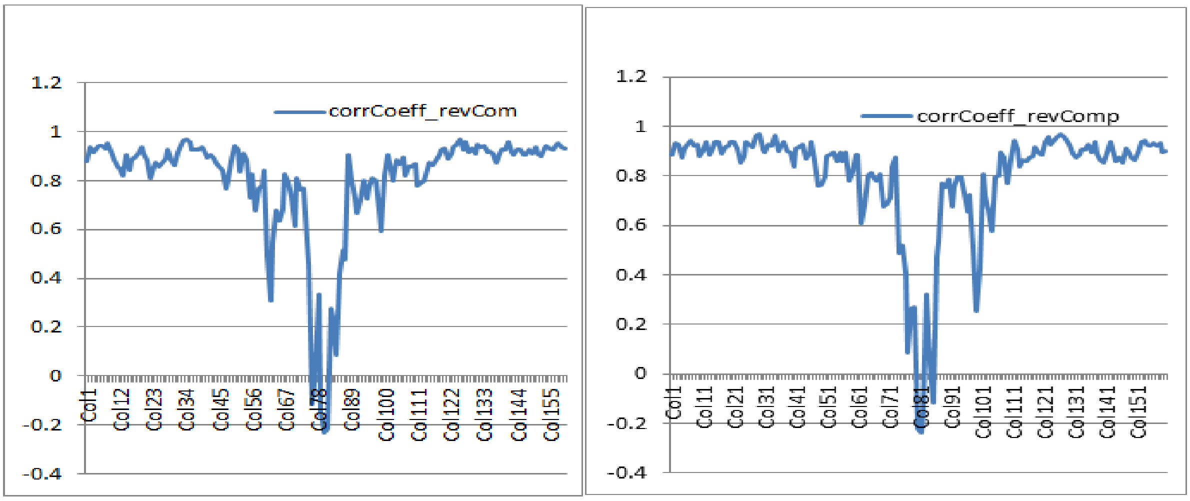


Figure 4-11: Correlation plot of di-nucleotide considering strandness of reverse complement

Figure shows a correlation plot of di-nucleotide at specific position with background distribution. (a) Reverse complement of negative strand (b) reverse complement of positive strand

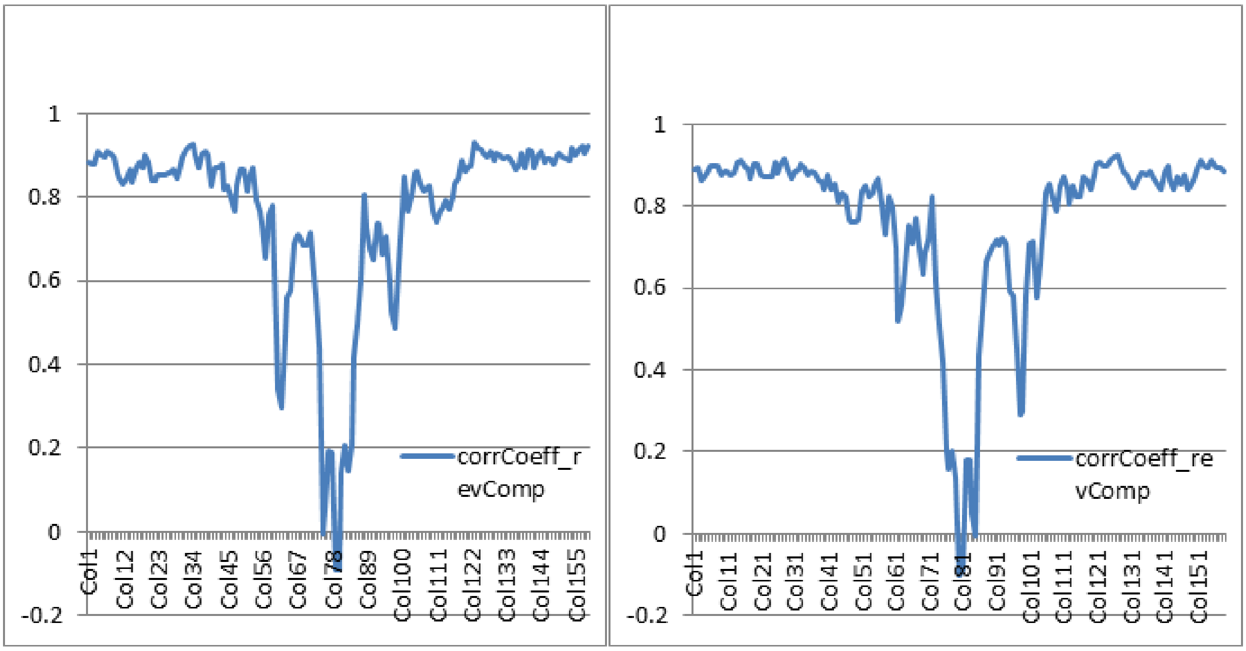


Figure 4-12: Correlation plot of tri-nucleotide considering strandness of reverse complement

Figure shows a correlation plot of tri-nucleotide at specific position with background distribution. (a) Reverse complement of negative strand (b) reverse complement of positive strand

The plot for reverse complement of positive strand demonstrates the mirror image of negative strand or vice versa. All the signals that were present in positive strand were also present in negative strand. However, these plots exhibited little variation than the original sequence and reverse complement sequence presented earlier. The reason is due to the discrimination in strand. The idea to make plots for strand specific was to check the dominance of signal in individual strand.

It was seen from the correlation plot of mononucleotide, di-nucleotide and tri-nucleotide that there is existence of one clear signal at 18 and 54 position plots. This was described by abrupt change in correlation value in the plot. From the sequence logo, there does not have existence of any significant conserved residue. However, a pattern of only Cs and only Gs were discontinued by base A forming a pair with T was clearly observed from logo in Figure 4-15. This region may play a vital role in transcription factor binding during transcription. From the logo of positive plot in Figure 4-15, it can be seen that motif lies from position 31 to 41(41 inclusive). Interestingly, it looks like the formation of hairpin (cruciform structure). Figure 4-16 gives the plot of true motif, Figure 4-13 and Figure 4-14 displays the logo for false_false positive and false positive respectively. The logo of positive set was quite different than that of false positive set and far more different than false_false positive set.

4.1.3 Logo Plot

The fasta format for positive and false positive region was used to make a logo. The logo for false_false positive sequence was also made. The generation of false_false positive region has been explained in earlier chapter. The plots were also made for the reverse complement of positive sequences, positive sequence and the true positive binding motifs. These entire logo plots were made using tool Seq2Logo [68]. Furthermore, logo plot for extended regions are presented in Appendix IV.

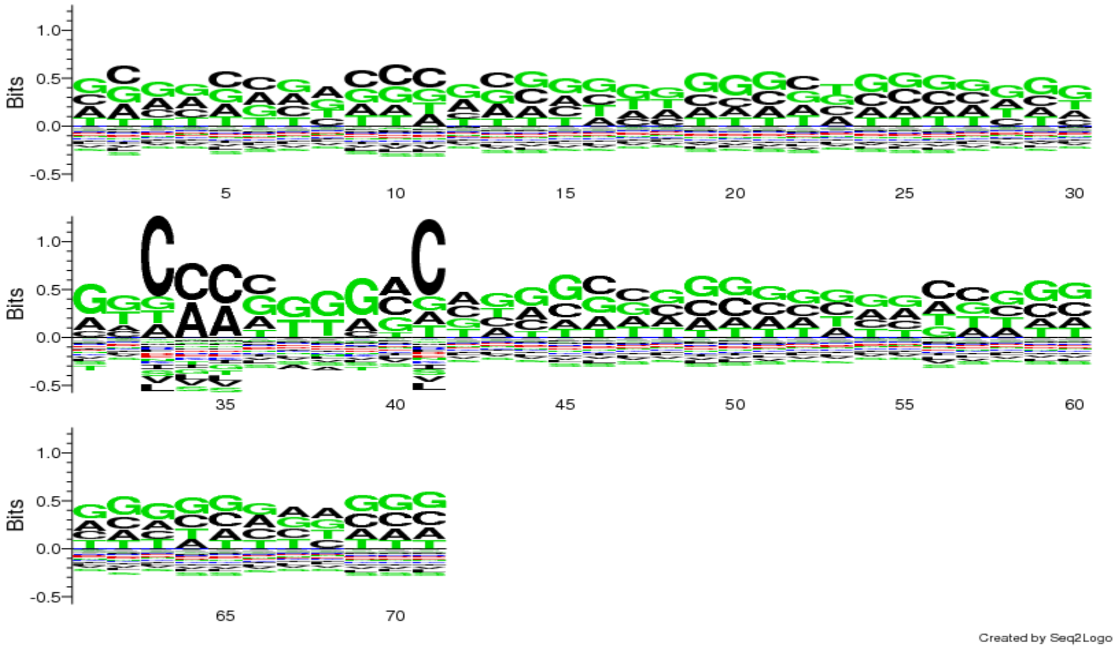


Figure 4-13: Logo of false_false positive sequence

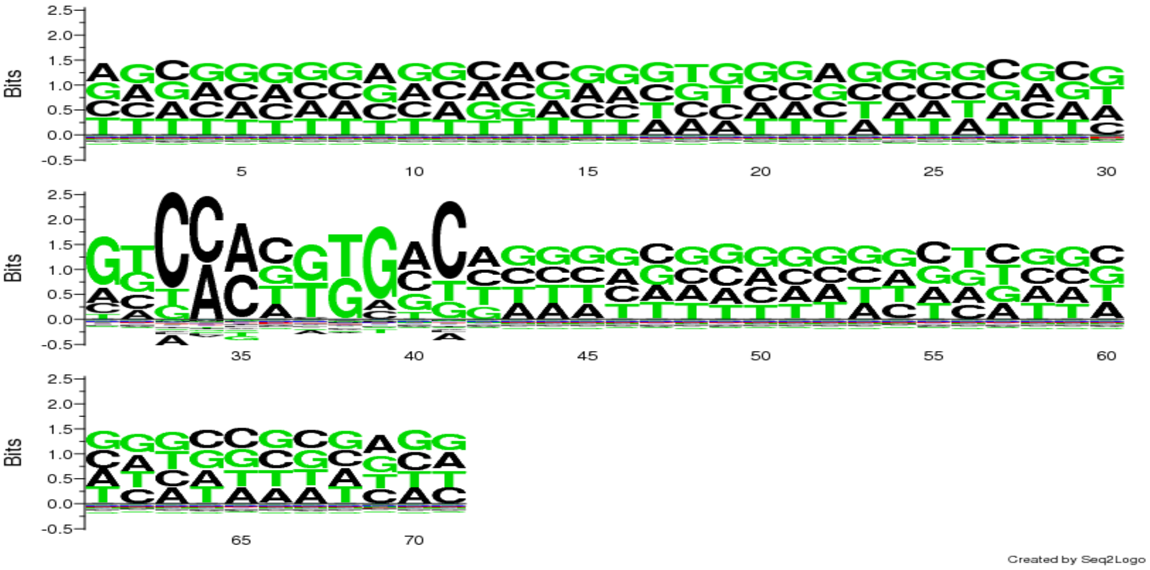


Figure 4-14: Logo of false positive sequence

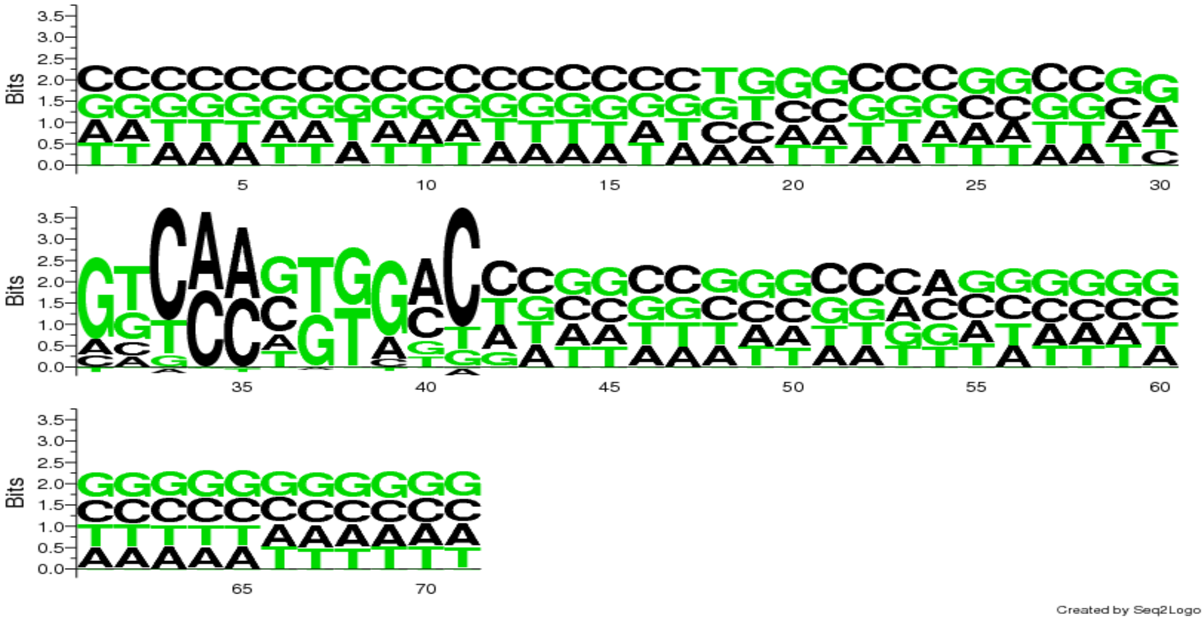


Figure 4-15: Logo of positive sequence

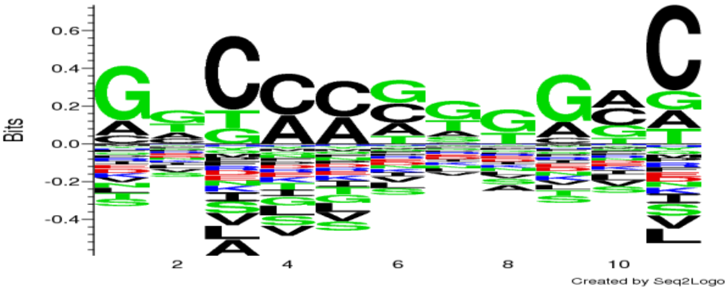


Figure 4-16: Logo of true binding motif

From the figure, it can be seen that the region for false_false positive was completely different from that of false positive and positive region. It can be seen that the positive regions has more conserved residues around binding site than that of false positive region. Interestingly, it can be seen that nucleotides around the region of binding sites were complement to each other. This gives the idea of formation of hairpin more precisely a cruciform structure. This structure formation is not present in false positive region which also provide a strong ground that the false region used in this work is good enough. Figure 4-16 shows a distribution of nucleotides in true binding motif. This logo was made by extracting the fasta region from the provided data of true positive binding sites.

The logo of reverse complement of positive sequences was made using Sequence2Logo. Figure 4-17 shows the plot for region 160 nts. The binding motif lies from76 th position to 86th position. This figure shows the similarity with logo plot of positive regions.

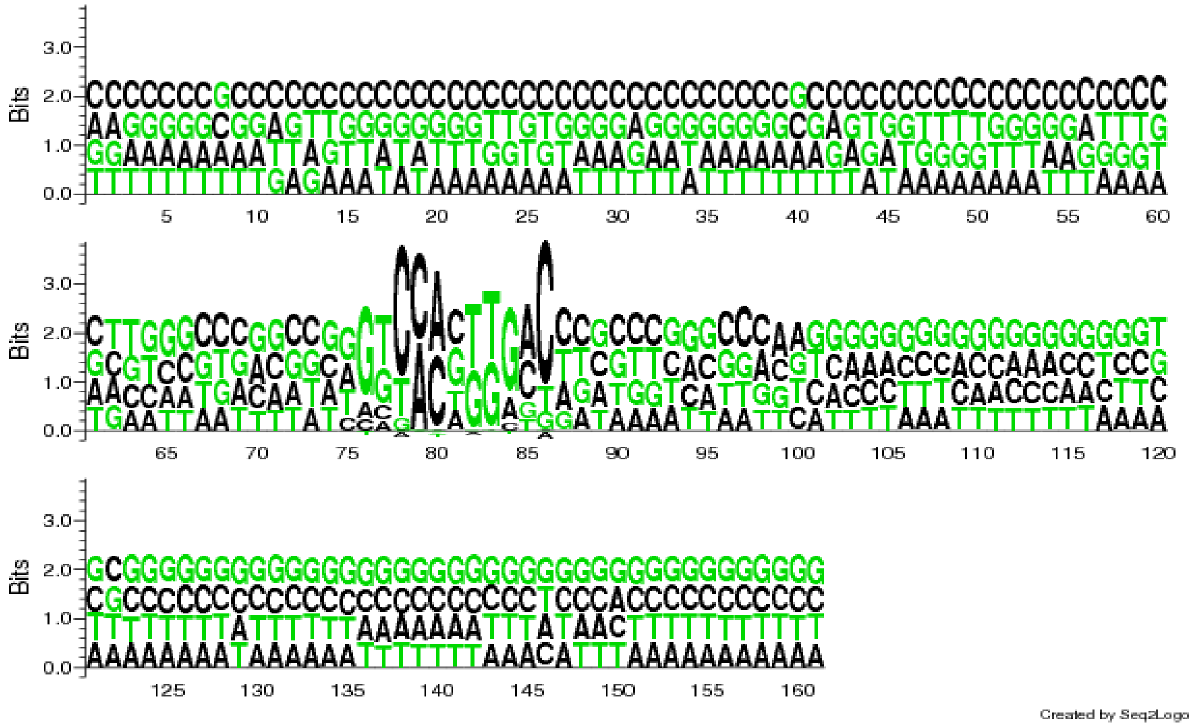


Figure 4-17: Logo plot of reverse complement of positive region

The logo plot for positive strand and negative strand were also made. These plots are shown in Figure 4-18 and Figure 4-19. Form the both figures it can be seen that the nucleotides are highly conserved in binding sites which is expected. The weight of nucleotides in conserved position was decreased when the plot is made without considering the polarity of strands.

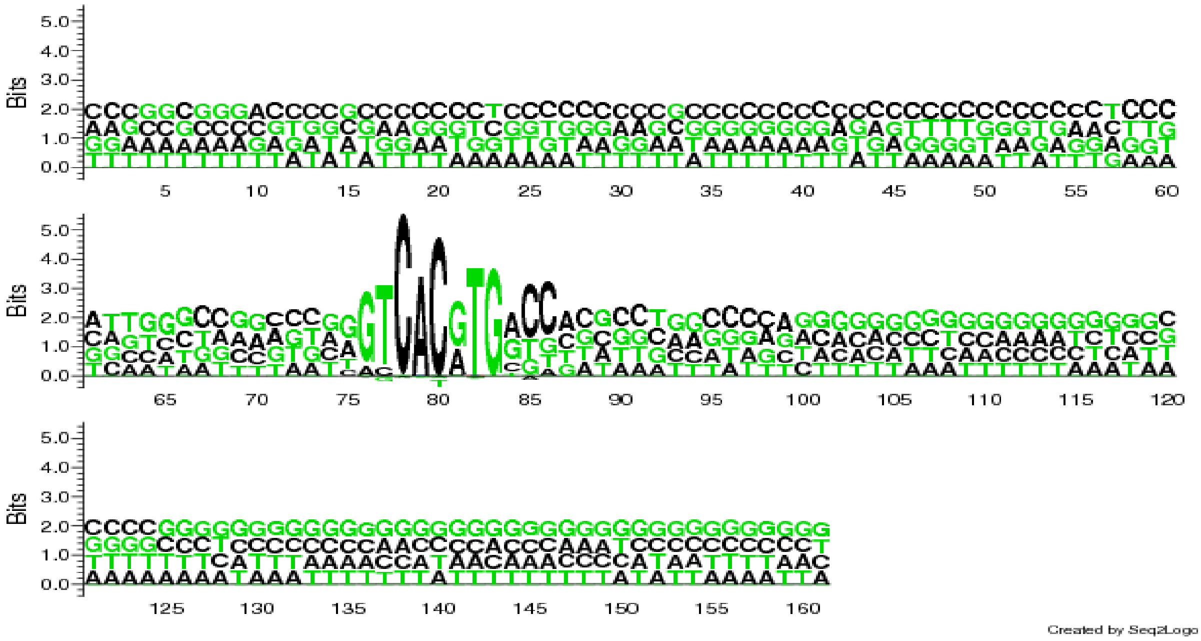


Figure 4-18: Logo plot of negative strand

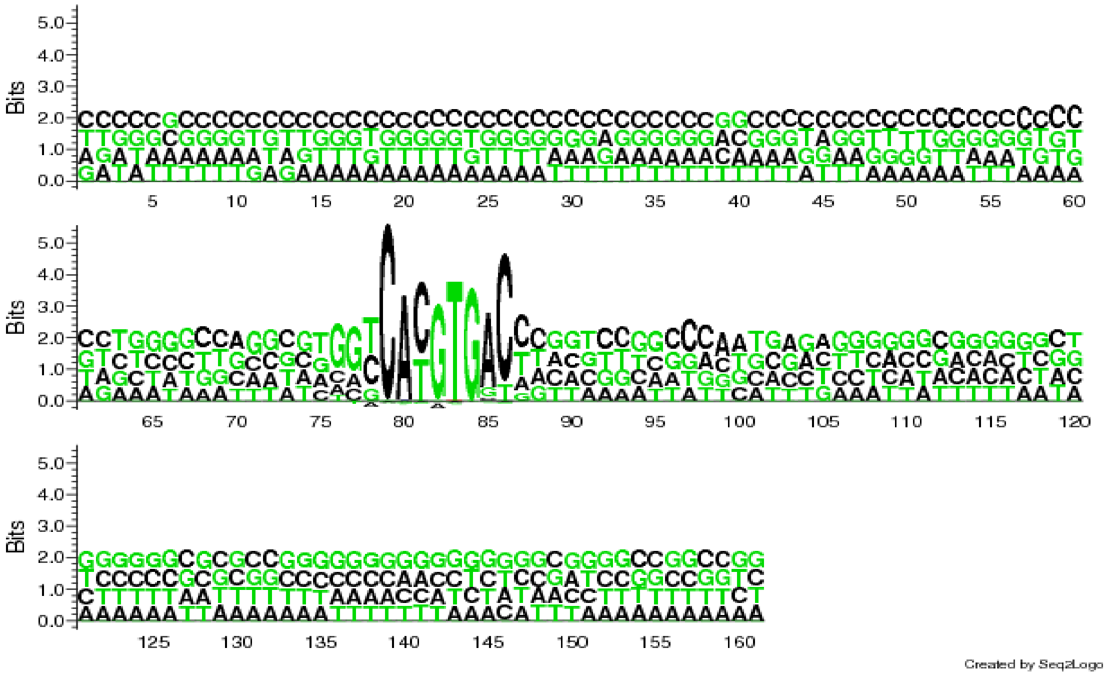


Figure 4-19: Logo plot of positive strand

4.1.4 FIMO output observation

The output of FIMO for false_false positive region contained the hits with higher p-vale than hits obtained for false positive region. Figure for these hits are presented in Appendix Since, hits were very few in numbers, these false_false positive set was not used in classification. However, it was used to compare the nature of composition of dataset against false positive dataset.

4.1.5 Binding region forming cruciform structure

The output of sequence logo from positive fasta region gives the possibility of cruciform structure around TFBS region. The structure can be drawn from the logo is shown in Figure 4-20.

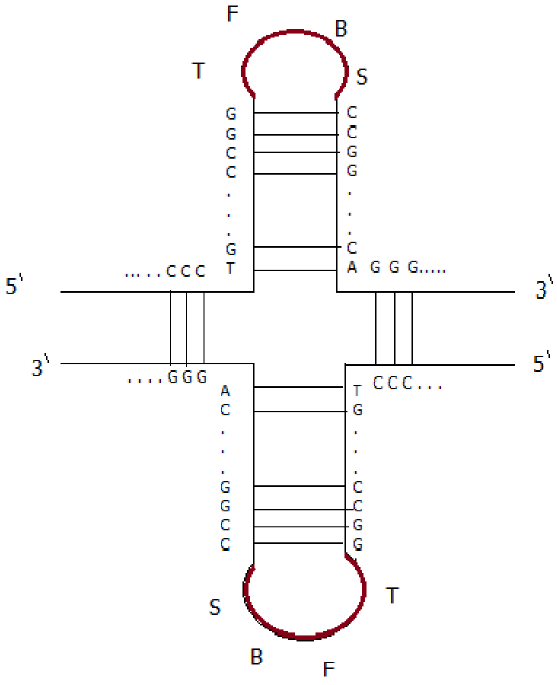


Figure 4-20: Possible cruciform structure

Figure shows a possible cruciform structure observed from the sequence logo of positive sequence, ref from Figure 4-15.

An attempt was made to test the TFBS forming the cruciform structure. The idea to identify cruciform structure computationally is described in background section and the algorithm developed to test the structure was presented in methodology. The high score was expected along the diagonal of the output matrix to confirm cruciform structure. However, the score obtained did not show any significant high score patterns along diagonal except in region around binding sites. This is shown in Figure 4-21.

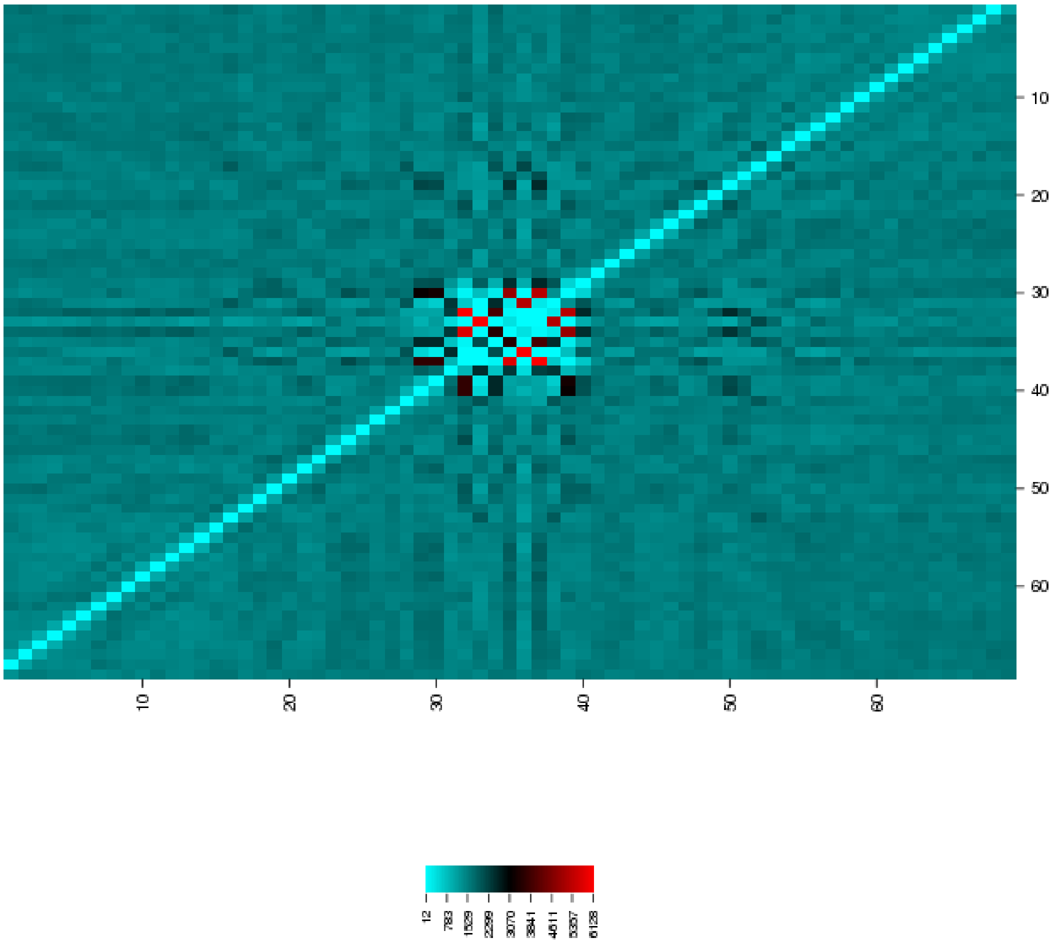


Figure 4-21: Heat Map

Figure 4-21 shows the heat map plot from the score obtained from the computation. The color shows the distribution of scores. Scores are represented by color coding as red color reflects the value with high score and the low score are shown in green. From the heat map, it can be observed that high scores are found around center of diagonal. This high score region is the binding region, which is shown by red and black spots. There is also presence of high score at offset plus one and minus one along the central diagonal. From the logo of true positive and appendix V, the motif can be estimated as GTCACGTGGCC. The reverse complement of this sequence will be GGCCACGTGAC. Suppose, similarity matrix is computed between this motif and reverse complement of it and is shown in figure 4-22.

	G	T	C	A	C	G	T	G	G	C	C
G	1					1		1	1		
G	1					1		1	1		
C			1		1						1
C			1		1						1
A				**1**							
C			1		**1**						1
G	1					1		1	1		
T		1					1				
G	1					1		1	1		
A				1							
C			1		1					1	1

Figure 4-22: Similarity Matrix of Motif and its reverse complement

From the matrix, M(3,6) and M(6,9) are high due to similarity with reverse complement. This scenario is reflected in figure 4-21 that is offset plus one and minus one along the central diagonal. Higher value along central diagonal signifies that there are lots of sequences where A-T pairs and G-C pairs are formed. These are also shown in logo plot of positive sequences in Figure 4-15 where base at position 33 and 34 is highly conserved as complementary pairs at positions 38 and 39 respectively.

4.1.6 Classification results

The classification output of Weka is shown in table. Here, I used SVM with three different kernels to classify. I also used Naïve Bayes and random forest method to compare the classification strategy and to estimate the optimum possible classification accuracy. I also tried to classify the vectors based upon standalone features like classification only with genomic features, only with physical properties and combining genomic and physical properties. It was done only to compare the effect of different included features. All the values are presented in three different tables.

Table 4-3: Classification result considering only genomic properties

Method	**TP**	**TN**	**FP**	**FN**	**Sp**	**Sn**	**F-score**	**Accuracy (%)**
SVM -poly Kernel	54	68	29	64	0.701	0.457	0.537	56.74
RBF	2	97	0	116	1	01	0. 03	46.04
PUK	94	55	42	24	0. 56	0. 79	0. 74	69.30
NB	32	86	11	86	0. 88	0. 27	0.394	54.88
RF	98	92	5	20	0. 948	0. 83	0. 88	88.37

Table 4-4: Classification result considering only physical properties

Method	**TP**	**TN**	**FP**	**FN**	**Sp**	**Sn**	**F-score**	**Accuracy (%)**
SVM -poly Kernel	43	70	27	75	0.721	0.364	0.457	52.55
RBF	58	64	33	60	0. 659	0. 491	0. 555	56.74
PUK	92	72	25	26	0. 742	0. 779	0. 782	76.27
NB	75	55	42	43	0. 567	0. 635	0. 638	60.46
RF	94	91	6	24	0.938	0. 796	0. 862	86.04

Table 4-5: Classification results considering all properties

This includes both genomic and physical properties, the optimum predictions are shown in bold text.

Method	**TP**	**TN**	**FP**	**FN**	**Sp**	**Sn**	**F-score**	**Accuracy (%)**
SVM-poly Kernel	46	72	25	72	0.742	0.389	0.551	54.87

RBF	56	68	29	62	0.701	0.474	0.573	57.67
PUK	**91**	**76**	**21**	**27**	**0.783**	**0.771**	**0.791**	**77.67**
NB	68	59	38	50	0.608	0.576	0.607	59.06
RF	**101**	**92**	**5**	**17**	**0.948**	**0.855**	**0.901**	**89.76**

From the result of classification, it can be seen that SVM with PUK kernel approximates best among all three SVM kernels with highest accuracy, sensitivity and specificity. Accordingly, random forest (RF) approximates exceptionally well among all different classification algorithms with highest accuracy, F1-value, sensitivity and specificity.

Comparing the result from table, it can be observed that classification with single feature either genomic or physical yields mix fair predictions. However, mixing this all features gives very good prediction. Another important conclusion that can be observed is the choice of classification strategy. In case of RAFT, PUK kernel gives the best optimal classification while using SVM. However, random forest strategy outperforms all other algorithms that were tested in all cases giving the highest accuracy of prediction.

4.1.7 Important features for classification

I tried to find the feature that has high influence during classification. It can be done by attribute select panel available in explorer interface of Weka. I used classifier subset evaluator as attribute evaluator. Then, best first method to classify for Random forest and Naïve Bayes was selected. This evaluator gives 5 and 4 important attributes for Naïve and RF approach respectively. The attribute evaluator used in previous case does not work for SVM; therefore, I selected SVM attribute evaluator with ranker method. This ranks all the features used in classification. The results observed are shown in

Table 4-6.

Table 4-6: Important properties for classification

These properties were observed from attribute select panel of weka explorer. The features are presented according to the order of importance. The common features are highlighted with bold and italic formats.

Naïve Bayes (best first)	Random Forest (best first)	SVM (based on rank)
skew_DNADenat	***GCskew***	TSSdistance
mean_DNADenat	**sd_BDNAtwist**	mean_ProtDNAtwist
skew_StackEng	*kurt_BDNAtwist*	***kurt_StackEng***
sd_StackEng	***mean_DNADenat***	**GCskew**
kurt_StackEng		sd_DNADenat
		sd_StackEng
		sd_BDNAtwist
		kurt_DNADenat
		mean_BDNAtwist

From the

Table 4-6, it was observed that some of the properties were commonly predicted as important properties by all three classification techniques. It can be seen that Naïve approach gives more importance to statistical features. However, RF and SVM shows genomic feature as the best feature. I tried to predict the accuracy with considering only these respective features shown by attribute select. The output for new dataset based on these important features is shown in Table 4-7.

Table 4-7: Classification result considering only important properties

Method	TP	TN	FP	FN	Sp	Sn	F-score	Accuracy (%)
NB	79	60	37	39	0. 618	0. 669	0. 675	64.65
PUK	87	72	25	31	0.742	0.737	0.756	73.95
RF	99	83	14	19	0. 855	0. 83	0.857	84.65

The accuracy using RF and PUK using only important features is lightly lowered than the maximum accuracy obtained using all attributes. However, classification accuracy of Naïve Bayes increased with considering only important features.

Chapter 5

5 Conclusions

RAFT attempted to classify real and false transcription factor binding sites based upon the genomic and structural properties of DNA strand. TF-specific ChIP-seq data was used to compute genomic and structural features. These features were used as attributes for property based classification of real and false transcription factor binding sites. Machine learning algorithms like Naïve Bayes, Random Forest and Support Vector Machines were used to classify different instances as negative and positive cases. Three different types of kernels were used for classification using SVM.

The result of classification showed that the RF outperforms all other tested algorithms in terms of accuracy, specificity and sensitivity. RF was able to classify with highest specificity and sensitivity of 0.948 and 0.855 respectively. The classification result of SVM was highly depended on the choice of kernel. SVM produces optimum accuracy of 77.67% with the use of PUK kernel. This figure of accuracy was far greater than the accuracy obtained by Naïve Bayes. SVM being useful and highly used in bioinformatics generates more misclassification than RF. Naïve Bayes algorithm does not fit in this type of classification since it has lowest specificity, sensitivity and accuracy in each tested case.

Importantly, the additional signals around transcription factor binding site were observed in correlation plot. The presence of these signals was also supported by the logo plot. The logo plot indicated the conserved region and the distribution of nucleotides. The correlation plot for dataset with different extended region gave the knowledge to use the proper region of sequences to compute the properties for classification. The correlation plot displayed almost flat plot except region around binding site.

Interestingly, a cruciform like structure was observed from the logo plot of positive region. The test was done to verify cruciform structure. But the result of the test did not verify the cruciform structure formation around the binding site as expected.

Finally, choice of features has a significant impact in the classification performance. For instance, RF showed only 4 important features that were significant for classification. In addition, Naïve approach showed increase performance by using only important properties for classification.

5.1 Limitation

This project is limited to the source of data that is obtained from TF-specific ChIP-seq data and the PWM motif for USF1. This project only focuses in the certain physical and genomic properties.

5.2 Future Work

This project can be further enhanced with addition of more genomic properties like k-mers and AT Skew. Addition of these features may increase the classification accuracy. The presence of additional signal that might play vital role in transcription can be studied. These signals may reveal more hidden secrets in transcription factor binding and transcription process. It may also be possible to extend this to cases where we do not have TF-specific ChIP-seq data, by identifying TF-independent properties from the first analysis, and use this for a more general classifier.

References

[1] Barabasi, A. L., & Oltvai, Z. N. (2004). Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, *5*(2), 101-113.

[2] Krylov, S. (2005). NECEEM for development, characterisation and analytical utilisation of apatmers. LabPlus International, Nov.

[3] Won, K. J., Ren, B., & Wang, W. (2010). Method Genome-wide prediction of transcription factor binding sites using an integrated model.

knowledge discovery, *2*(2), 121-167.

[4] Claverie, J. M., & Audic, S. (1996). The statistical significance of nucleotide position-weight matrix matches. *Computer applications in the biosciences: CABIOS*, *12*(5), 431-439.

[5] Wasserman, W. W., & Sandelin, A. (2004). Applied bioinformatics for the identification of regulatory elements. *Nature Reviews Genetics*, *5*(4), 276-287.

[6] Kulakovskiy, I., Levitsky, V., Oshchepkov, D., Bryzgalov, L., Vorontsov, I., & Makeev, V. (2013). From binding motifs in ChIP-seq data to improved models of transcription factor binding sites. *Journal of bioinformatics and computational biology*, *11*(01).

[7] Berman, B. P., Nibu, Y., Pfeiffer, B. D., Tomancak, P., Celniker, S. E., Levine, M., & Eisen, M. B. (2002). Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the Drosophila genome. *Proceedings of the National Academy of Sciences*, *99*(2), 757-762.

[8] Harbison, C. T., Gordon, D. B., Lee, T. I., Rinaldi, N. J., Macisaac, K. D., Danford, T. W.,& Young, R. A. (2004). Transcriptional regulatory code of a eukaryotic genome. *Nature*, *431*(7004), 99-104.

[9] Vaquerizas, J. M., Kummerfeld, S. K., Teichmann, S. A., & Luscombe, N. M. (2009). A census of human transcription factors: function, expression and evolution. *Nature Reviews Genetics*, *10*(4), 252-263.

[10] Maston, G. A., Evans, S. K., & Green, M. R. (2006). Transcriptional regulatory elements in the human genome. *Annu. Rev. Genomics Hum. Genet.*, *7*, 29-59.

[11] Lettice, L. A., Heaney, S. J., Purdie, L. A., Li, L., de Beer, P., Oostra, B. A., ... & de Graaff, E. (2003). A long-range Shh enhancer regulates expression in the developing limb and fin and is associated with preaxial polydactyly. *Human molecular genetics*, *12*(14), 1725-1735.

[12] Berta, P., Hawkins, J. B., Sinclair, A. H., Taylor, A., Griffiths, B. L., Goodfellow, P. N., & Fellous, M. (1990). Genetic evidence equating SRY and the testis-determining factor. *Nature*, *348*(6300), 448-450.

[13] Shamovsky, I., & Nudler, E. (2008). New insights into the mechanism of heat shock response activation. *Cellular and Molecular Life Sciences*, *65*(6), 855-861.

[14] Benizri, E., Ginouves, A., & Berra, E. (2008). The magic of the hypoxia-signaling cascade. *Cellular and molecular life sciences*, *65*(7-8), 1133-1149.

[15] Wheaton, K., Atadja, P., & Riabowol, K. (1996). Regulation of transcription factor activity during cellular aging. *Biochemistry and cell biology*, *74*(4), 523-534.

[16] Evan, G., Harrington, E., Fanidi, A., Land, H., Amati, B., & Bennett, M. (1994). Integrated control of cell proliferation and cell death by the c-myc oncogene.*Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, *345*(1313), 269-275.

[17] Bauer, A. L., Hlavacek, W. S., Unkefer, P. J., & Mu, F. (2010). Using sequence-specific chemical and structural properties of DNA to predict transcription factor binding sites. *PLoS computational biology*, *6*(11), e1001007.

[18] Lecture notes, University of Georgia, Regulatory transcription factors, retrieved on March 7, 2014, from http://www.bmb.uga.edu/mterns/bcmb8020/lectures/ppts/009%208020%20TF%20Domains%20.pdf

[19] Van Holde, K., & Zlatanova, J. (1994). Unusual DNA structures, chromatin and transcription. *Bioessays*, *16*(1), 59-68.

[20] Mikheikin, A. L., Lushnikov, A. Y., & Lyubchenko, Y. L. (2006). Effect of DNA supercoiling on the geometry of holliday junctions. *Biochemistry*, *45*(43), 12998-13006.

[21] Lyubchenko, Y. L., & Shlyakhtenko, L. S. (1997). Visualization of supercoiled DNA with atomic force microscopy in situ. *Proceedings of the National Academy of Sciences*, *94*(2), 496-501.

[22] Panayotatos, N., & Fontaine, A. (1987). A native cruciform DNA structure probed in bacteria by recombinant T7 endonuclease. *Journal of Biological Chemistry*, *262*(23), 11364-11368.

[23] Yahyaoui, W., Callejo, M., Price, G. B., & Zannis-Hadjopoulos, M. (2007). Deletion of the cruciform binding domain in CBP/14-3-3 displays reduced origin binding and initiation of DNA replication in budding yeast. *BMC molecular biology*, *8*(1), 27.

[24] Brázda, V., Laister, R. C., Jagelská, E. B., & Arrowsmith, C. (2011). Cruciform structures are a common DNA feature important for regulating biological processes. *BMC molecular biology*, *12*(1), 33.

[25] Poulet, A., Buisson, R., Faivre-Moskalenko, C., Koelblen, M., Amiard, S., Montel, F., ... & Giraud-Panis, M. J. (2009). TRF2 promotes, remodels and protects telomeric Holliday junctions. *The EMBO journal*, *28*(6), 641-651.

[26] Ogawa, N., & Biggin, M. D. (2012). High-throughput SELEX determination of DNA sequences bound by transcription factors in vitro. In *Gene Regulatory Networks* (pp. 51-63). Humana Press.

[27] Bulyk, M. L. (2004). Computational prediction of transcription-factor binding site locations. *Genome biology*, *5*(1), 201-201.

[28] Stephanie M, Modified Nucleoside Triphosphate Applications: An Overview of the SELEX Process, Retrieved on May 19, 2014, from http://www.trilinkbiotech.com/tech/selex.asp

[29] Stoltenburg, R., Reinemann, C., & Strehlitz, B. (2007). SELEX—a (r) evolutionary method to generate high-affinity nucleic acid ligands. *Biomolecular engineering*, *24*(4), 381-403.

[30] Pillai, S., & Chellappan, S. P. (2009). ChIP on chip assays: genome-wide analysis of transcription factor binding and histone modifications. In *Chromatin Protocols* (pp. 341-366). Humana Press.

[31] Liu, E. T., Pott, S., & Huss, M. (2010). Q&A: ChIP-seq technologies and the study of gene regulation. *BMC biology*, *8*(1), 56.

[32] Gershenzon, N. I., Stormo, G. D., & Ioshikhes, I. P. (2005). Computational technique for improvement of the position-weight matrices for the DNA/protein binding sites. *Nucleic acids research*, *33*(7), 2290-2301.

[33] Kulakovskiy, I., Levitsky, V., Oshchepkov, D., Bryzgalov, L., Vorontsov, I., & Makeev, V. (2013). From binding motifs in ChIP-seq data to improved models of transcription factor binding sites. *Journal of bioinformatics and computational biology*, *11*(01).

[34] Lingner, T., Aßhauer, K. P., Schreiber, F., & Meinicke, P. (2011). CoMet—a web server for comparative functional profiling of metagenomes. *Nucleic acids research*, *39*(suppl 2), W518-W523.

[35] Frith, M. C., Li, M. C., & Weng, Z. (2003). Cluster-Buster: Finding dense clusters of motifs in DNA sequences. *Nucleic acids research*, *31*(13), 3666-3668.

[36] Van Loo, P., Aerts, S., Thienpont, B., De Moor, B., Moreau, Y., & Marynen, P. (2008). ModuleMiner-improved computational detection of cis-regulatory modules: are there different modes of gene regulation in embryonic development and adult tissues. *Genome Biol*, *9*(4), R66.

[37] Won, K. J., Ren, B., & Wang, W. (2010). Method Genome-wide prediction of transcription factor binding sites using an integrated model.

[38] Mittag, F., Büchel, F., Saad, M., Jahn, A., Schulte, C., Bochdanovits, Z., ... & Sharma, M. (2012). Use of support vector machines for disease risk prediction in genome-wide association studies: Concerns and opportunities. *Human mutation*, *33*(12), 1708-1718.

[39] Johansson, P., & Ringnér, M. (2007). Classification of genomic and proteomic data using support vector machines. In *Fundamentals of Data Mining in Genomics and Proteomics* (pp. 187-202). Springer US.

[40] Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, *46*(1-3), 389-422.

[41] KR, S. (2011). Microarray Data Classification Using Support Vector Machine.*International Journal of Biometrics and Bioinformatics (IJBB)*, *5*(1), 10

[42] Nassif, H., Al-Ali, H., Khuri, S., & Keirouz, W. (2009). Prediction of protein-glucose binding sites using support vector machines. *Proteins: Structure, Function, and Bioinformatics*, *77*(1), 121-132.

[43] Kumar, M., Gromiha, M. M., & Raghava, G. P. S. (2008). Prediction of RNA binding sites in a protein using SVM and PSSM profile. *Proteins: Structure, Function, and Bioinformatics*, *71*(1), 189-194.

[44] Holloway, D. T., Kon, M., & DeLisi, C. (2005). Integrating genomic data to predict transcription factor binding. *Genome Informatics Series*, *16*(1), 83.

[45] Sun, Y., Robinson, M., Adams, R., Rust, A., & Davey, N. (2008). Prediction of Binding Sites in the Mouse Genome Using Support Vector Machines. In*Artificial Neural Networks-ICANN 2008* (pp. 91-100). Springer Berlin Heidelberg.

[46] Mukherjee, K., Abhipriya, A. S. V., & Pandey, D. M. (2013). SVM based model generation for binding site prediction on helix turn helix motif type of transcription factors in eukaryotes. *Bioinformation*, *9*(10), 500.

[47] Maienschein-Cline, M., Dinner, A. R., Hlavacek, W. S., & Mu, F. (2012). Improved predictions of transcription factor binding sites using physicochemical features of DNA. *Nucleic acids research*, *40*(22), e175-e175.

[48] Johnson, D. S., Mortazavi, A., Myers, R. M., & Wold, B. (2007). Genome-wide mapping of in vivo protein-DNA interactions. *Science*, *316*(5830), 1497-1502.

[49] Wang, X., & Zhang, X. (2011). Pinpointing transcription factor binding sites from ChIP-seq data with SeqSite. *BMC systems biology*, *5*(Suppl 2), S3.

[50] Wasserman, W. W., & Sandelin, A. (2004). Applied bioinformatics for the identification of regulatory elements. *Nature Reviews Genetics*, *5*(4), 276-287.

[51] Florquin, K., Saeys, Y., Degroeve, S., Rouze, P., & Van de Peer, Y. (2005). *Nucleic acids research*, *33*(13), 4255-4264.

[52] Ornstein, R. L., Rein, R., Breen, D. L., & Macelroy, R. D. (1978). An optimized potential function for the calculation of nucleic acid interaction energies I. Base stacking. *Biopolymers*, *17*(10), 2341-2360.

[53] El Hassan, M. A., & Calladine, C. R. (1996). Propeller-twisting of base-pairs and the conformational mobility of dinucleotide steps in DNA. *Journal of molecular biology*, *259*(1), 95-103.

[54] Olson, W. K., Gorin, A. A., Lu, X. J., Hock, L. M., & Zhurkin, V. B. (1998). DNA sequence-dependent deformability deduced from protein–DNA crystal complexes. *Proceedings of the National Academy of Sciences*, *95*(19), 11163-11168.

[55] Breslauer, K. J., Frank, R., Blöcker, H., & Marky, L. A. (1986). Predicting DNA duplex stability from the base sequence. *Proceedings of the National Academy of Sciences*, *83*(11), 3746-3750.

[56] Petra C S., (2008), Supervise classification of gene regulatory regions in the human genome. Thesis report, NTNU-Trondheim, 34-35

[57] Liao, G. C., Rehm, E. J., & Rubin, G. M. (2000). Insertion site preferences of the P transposable element in Drosophila melanogaster. *Proceedings of the National Academy of Sciences*, *97*(7), 3347-3351.

[58] Wikipedia, Kurtosis, Retrieved on June 14, 2014, from

http://en.wikipedia.org/wiki/Kurtosis

[59] Teschendorff, A. E., Naderi, A., Barbosa-Morais, N. L., & Caldas, C. (2006). PACK: Profile Analysis using Clustering and Kurtosis to find molecular classifiers in cancer. *Bioinformatics*, *22*(18), 2269-2275.

[60] NIST/SEMATECH, Measures of Skewness and kurtosis, Retrieved on June 4, 2014, from

http://www.itl.nist.gov/div898/handbook/eda/section3/eda35b.htm

[61]Wikipedia, Skewness, Retrieved on June 14, 2014, from
http://en.wikipedia.org/wiki/Skewness

[62] Deaton, A. M., & Bird, A. (2011). CpG islands and the regulation of transcription. *Genes & development*, *25*(10), 1010-1022.

[63] Yean, D., & Gralla, J. (1996). Transcription activation by GC-boxes: evaluation of kinetic and equilibrium contributions. *Nucleic acids research*, *24*(14), 2723-2729.

[64] Arakawa, K., & Tomita, M. (2006). The GC skew index: a measure of genomic compositional asymmetry and the degree of replicational selection. *Evolutionary bioinformatics online*, *3*, 159-168.

[65] Fasta format description, Retrieved on June 2, 2014, from

http://www.bioinformatics.nl/tools/crab_fasta.html

[66] Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization.

[67] Frank, E., Hall, M., Trigg, L., Holmes, G., & Witten, I. H. (2004). Data mining in bioinformatics using Weka. *Bioinformatics, 20*(15), 2479-2481.

[68] Lee, C. (2014). Machine Learning Approaches to Transcription Factor Binding Site Search and Visualization.

[69] Vapnik, V. (2000). *The nature of statistical learning theory*. springer.

[70] Bousquet, O., Boucheron, S., & Lugosi, G. (2004). Introduction to statistical learning theory. In *Advanced Lectures on Machine Learning* (pp. 169-207). Springer Berlin Heidelberg.

[71] Bennett, K. P., & Campbell, C. (2000). Support vector machines: hype or hallelujah?. *ACM SIGKDD Explorations Newsletter*, *2*(2), 1-13.

[72] Wikipedia, Support vector machines, Reterived on 16 March 2014, from

http://en.wikipedia.org/wiki/Support_vector_machine

[73] Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, *2*(2), 121-167.

[74] Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992, July). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory* (pp. 144-152). ACM.

[75] Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

[76] Liaw, A., & Wiener, M. (2002). Classification and Regression by randomForest.*R news*, *2*(3), 18-22.

[77] Díaz-Uriarte, R., & De Andres, S. A. (2006). Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, *7*(1), 3.

[78] Moorthy, K., & Mohamad, M. S. (2011). Random forest for gene selection and microarray data classification. *Bioinformation*, *7*(3), 142.

[79] Anaissi, A., Kennedy, P. J., Goyal, M., & Catchpoole, D. R. (2013). A balanced iterative random forest for gene selection from microarray data. *BMC bioinformatics*, *14*(1).

[80] Notes, Naïve-Bayes classification algorithm, Retrieved June 10, 2014, from

http://software.ucv.ro/~cmihaescu/ro/teaching/AIR/docs/Lab4-NaiveBayes.pdf

[81] Rosen, G. L., Reichenberger, E. R., & Rosenfeld, A. M. (2011). NBC: the Naive Bayes Classification tool webserver for taxonomic classification of metagenomic reads. *Bioinformatics*, *27*(1), 127-129.

[82] Sambo, F., Trifoglio, E., Di Camillo, B., Toffolo, G. M., & Cobelli, C. (2012). Bag of Naïve Bayes: biomarker selection and classification from genome-wide SNP data. *BMC bioinformatics*, *13*(Suppl 14), S2.

[83] Tim B, Java Source Code, reterived on Feb 19, 2014, from

http://www.javadocexamples.com/java_source/com/discursive/jccook/math/StatExample.java.html

[84] Quinlan, A. R., & Hall, I. M. (2010). BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, *26*(6), 841-842.

[85] Bailey, T. L., Boden, M., Buske, F. A., Frith, M., Grant, C. E., Clementi, L., ... & Noble, W. S. (2009). MEME SUITE: tools for motif discovery and searching.*Nucleic acids research*, gkp335.

[86] The MEME Suite, Online material, Retrieved on March 17, 2013, from , http://meme.nbcr.net/meme/intro.html.

[87] Grant, C. E., Bailey, T. L., & Noble, W. S. (2011). FIMO: scanning for occurrences of a given motif. *Bioinformatics*, *27*(7), 1017-1018.

[88] Thomsen, M. C. F., & Nielsen, M. (2012). Seq2Logo: a method for construction and visualization of amino acid binding motifs and sequence profiles including sequence weighting, pseudo counts and two-sided representation of amino acid enrichment and depletion. *Nucleic acids research*, *40*(W1), W281-W287

[89] Heat Map online tool, Retrieved on 07 July 2014, from

http://discover.nci.nih.gov/cimminer/h

Appendices

Appendix I

Correlation matrix for different dinucleotide properties, red color represents the property with least absolute correlation; green represents the highest absolute correlation. In this analysis the score with only 3 digits after decimal was considered. Naming is used in short form of structural properties discussed in background section.

	STACK energy	**prop twist**	**protin deform**	**free eng**	**Disrupt eng**	**DNA denat**	**BDNA twist**	**ProtDNA twist**	**Stab eng ZDNA**
STACK energy	1	-0,288	**0,043**	0,568	-0,351	-0,889	**-0,174**	0,228	**-0,159**
prop twist	-0,288	1	0,667	-0,686	0,556	0,289	**-0,046**	**0,162**	-0,637
protin deform	**0,043**	0,667	1	-0,574	0,574	-0,224	**-0,033**	0,683	-0,918
free eng	0,568	-0,686	-0,574	1	-0,907	-0,580	**0,165**	**-0,070**	0,570
Disrupt eng	-0,351	0,556	0,574	-0,907	1	0,402	**-0,159**	**0,205**	-0,588
DNA denat	-0,889	0,289	**-0,224**	-0,580	0,402	1	**-0,067**	-0,515	0,253
BDNA twist	-0,174	-0,047	-0,033	0,165	-0,159	-0,067	1	0,03	0,280
ProtDNA twist	0,228	0,162	0,683	-0,070	0,205	-0,515	0,037	1	-0,700
Stab eng ZDNA	-0,159	-0,637	-0,918	0,570	-0,588	0,253	0,280	-0,70	1
SUM	-0,023	1,017	1,217	-0,515	0,732	-0,331	1,002	1,030	-0,899
abs average	**0,411**	0,481	**0,524**	**0,569**	**0,527**	**0,46**	**0,218**	**0,400**	0,567

Appendix II

Possible di-nucleotides and tri-nucleotides combination

Table II-A: di-nucleotides

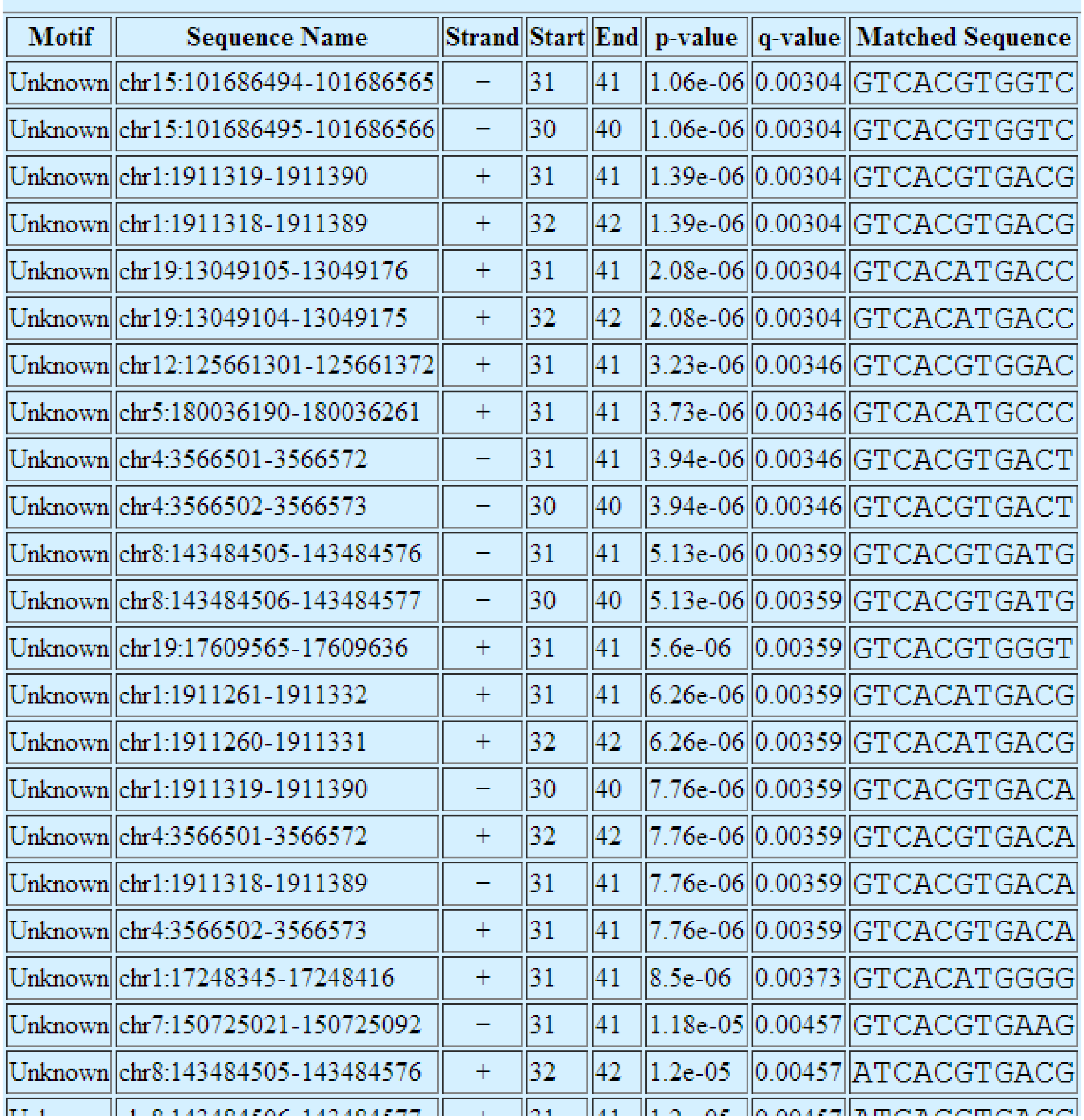
SN	Di-nucleotides	SN	Di-nucleotides
1	AA	9	CT
2	CC	10	CG
3	TT	11	TA
4	GG	12	TC
5	AC	13	TG
6	AT	14	GA
7	AG	15	GC
8	CA	16	GT

Table II-B: Tri-nucleotides

SN	Tri-nucl	SN	Tri-nucl	SN	Tri-nucl	SN	Tri-nucl
1	AAA	17	CAA	33	GAA	49	TAA
2	AAC	18	CAC	34	GAC	50	TAC
3	AAG	19	CAG	35	GAG	51	TAG
4	AAT	20	CAT	36	GAT	52	TAT
5	ACA	21	CCA	37	GCA	53	TCA
6	ACC	22	CCC	38	GCC	54	TCC
7	ACG	23	CCG	39	GCG	55	TCG
8	ACT	24	CCT	40	GCT	56	TCT
9	AGA	25	CGA	41	GGA	57	TGA
10	AGC	26	CGC	42	GGC	58	TGC
11	AGG	27	CGG	43	GGG	59	TGG
12	AGT	28	CGT	44	GGT	60	TGT
13	ATA	29	CTA	45	GTA	61	TTA
14	ATC	30	CTC	46	GTC	62	TTC
15	ATG	31	CTG	47	GTG	63	TTG
16	ATT	32	CTT	48	GTT	64	TTT

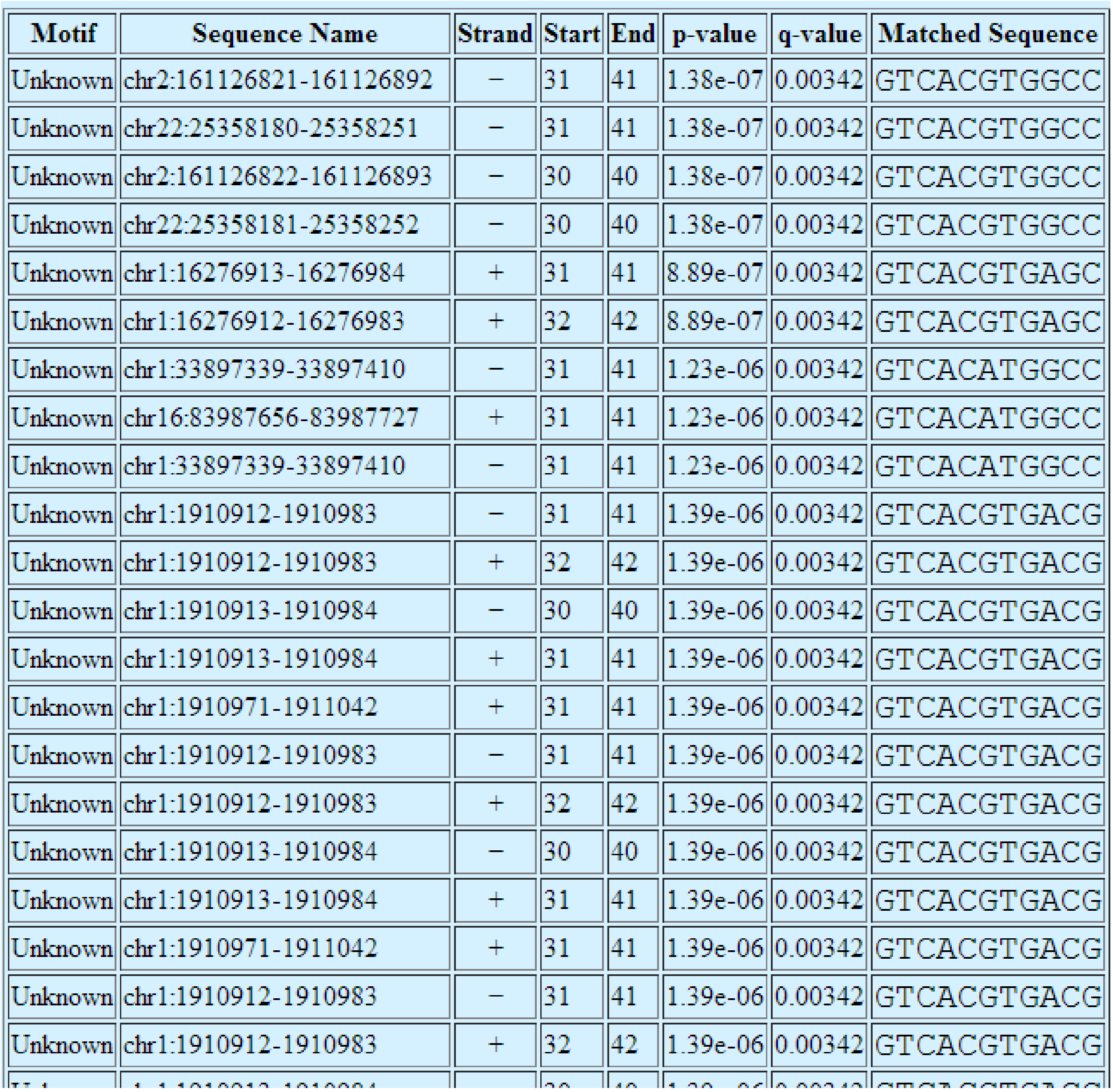
Appendix III

Sample of FIMO output of positive, false positive and false_false positive set



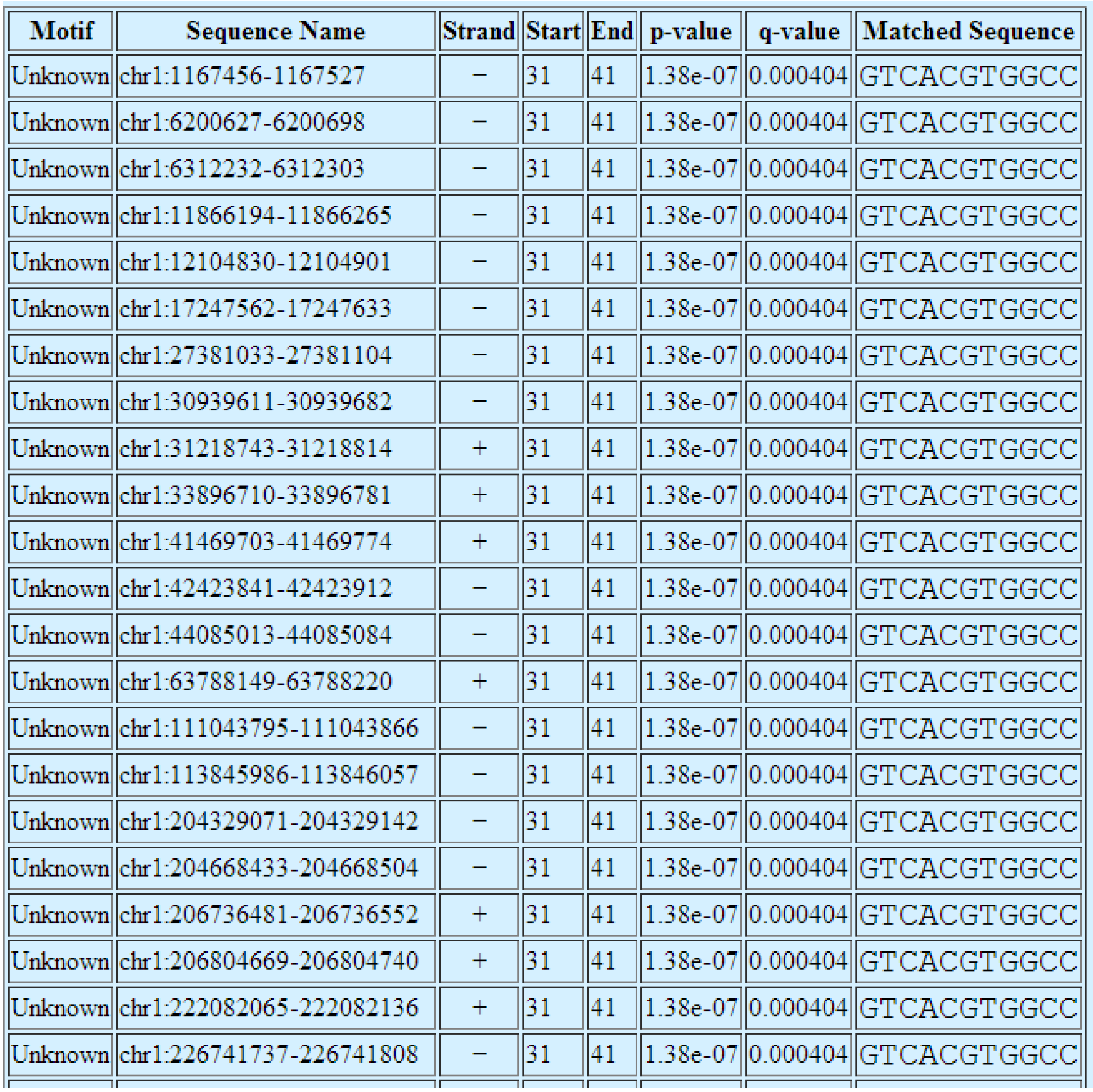
Motif	Sequence Name	Strand	Start	End	p-value	q-value	Matched Sequence
Unknown	chr15:101686494-101686565	−	31	41	1.06e-06	0.00304	GTCACGTGGTC
Unknown	chr15:101686495-101686566	−	30	40	1.06e-06	0.00304	GTCACGTGGTC
Unknown	chr1:1911319-1911390	+	31	41	1.39e-06	0.00304	GTCACGTGACG
Unknown	chr1:1911318-1911389	+	32	42	1.39e-06	0.00304	GTCACGTGACG
Unknown	chr19:13049105-13049176	+	31	41	2.08e-06	0.00304	GTCACATGACC
Unknown	chr19:13049104-13049175	+	32	42	2.08e-06	0.00304	GTCACATGACC
Unknown	chr12:125661301-125661372	+	31	41	3.23e-06	0.00346	GTCACGTGGAC
Unknown	chr5:180036190-180036261	+	31	41	3.73e-06	0.00346	GTCACATGCCC
Unknown	chr4:3566501-3566572	−	31	41	3.94e-06	0.00346	GTCACGTGACT
Unknown	chr4:3566502-3566573	−	30	40	3.94e-06	0.00346	GTCACGTGACT
Unknown	chr8:143484505-143484576	−	31	41	5.13e-06	0.00359	GTCACGTGATG
Unknown	chr8:143484506-143484577	−	30	40	5.13e-06	0.00359	GTCACGTGATG
Unknown	chr19:17609565-17609636	+	31	41	5.6e-06	0.00359	GTCACGTGGGT
Unknown	chr1:1911261-1911332	+	31	41	6.26e-06	0.00359	GTCACATGACG
Unknown	chr1:1911260-1911331	+	32	42	6.26e-06	0.00359	GTCACATGACG
Unknown	chr1:1911319-1911390	−	30	40	7.76e-06	0.00359	GTCACGTGACA
Unknown	chr4:3566501-3566572	+	32	42	7.76e-06	0.00359	GTCACGTGACA
Unknown	chr1:1911318-1911389	−	31	41	7.76e-06	0.00359	GTCACGTGACA
Unknown	chr4:3566502-3566573	+	31	41	7.76e-06	0.00359	GTCACGTGACA
Unknown	chr1:17248345-17248416	+	31	41	8.5e-06	0.00373	GTCACATGGGG
Unknown	chr7:150725021-150725092	−	31	41	1.18e-05	0.00457	GTCACGTGAAG
Unknown	chr8:143484505-143484576	+	32	42	1.2e-05	0.00457	ATCACGTGACG

Fig III-A: FIMO output for false_false positive region after aligning binding site and extending to 30 nts both sides. Here only few hits are shown.



Motif	Sequence Name	Strand	Start	End	p-value	q-value	Matched Sequence
Unknown	chr2:161126821-161126892	−	31	41	1.38e-07	0.00342	GTCACGTGGCC
Unknown	chr22:25358180-25358251	−	31	41	1.38e-07	0.00342	GTCACGTGGCC
Unknown	chr2:161126822-161126893	−	30	40	1.38e-07	0.00342	GTCACGTGGCC
Unknown	chr22:25358181-25358252	−	30	40	1.38e-07	0.00342	GTCACGTGGCC
Unknown	chr1:16276913-16276984	+	31	41	8.89e-07	0.00342	GTCACGTGAGC
Unknown	chr1:16276912-16276983	+	32	42	8.89e-07	0.00342	GTCACGTGAGC
Unknown	chr1:33897339-33897410	−	31	41	1.23e-06	0.00342	GTCACATGGCC
Unknown	chr16:83987656-83987727	+	31	41	1.23e-06	0.00342	GTCACATGGCC
Unknown	chr1:33897339-33897410	−	31	41	1.23e-06	0.00342	GTCACATGGCC
Unknown	chr1:1910912-1910983	−	31	41	1.39e-06	0.00342	GTCACGTGACG
Unknown	chr1:1910912-1910983	+	32	42	1.39e-06	0.00342	GTCACGTGACG
Unknown	chr1:1910913-1910984	−	30	40	1.39e-06	0.00342	GTCACGTGACG
Unknown	chr1:1910913-1910984	+	31	41	1.39e-06	0.00342	GTCACGTGACG
Unknown	chr1:1910971-1911042	+	31	41	1.39e-06	0.00342	GTCACGTGACG
Unknown	chr1:1910912-1910983	−	31	41	1.39e-06	0.00342	GTCACGTGACG
Unknown	chr1:1910912-1910983	+	32	42	1.39e-06	0.00342	GTCACGTGACG
Unknown	chr1:1910913-1910984	−	30	40	1.39e-06	0.00342	GTCACGTGACG
Unknown	chr1:1910913-1910984	+	31	41	1.39e-06	0.00342	GTCACGTGACG
Unknown	chr1:1910971-1911042	+	31	41	1.39e-06	0.00342	GTCACGTGACG
Unknown	chr1:1910912-1910983	−	31	41	1.39e-06	0.00342	GTCACGTGACG
Unknown	chr1:1910912-1910983	+	32	42	1.39e-06	0.00342	GTCACGTGACG

Fig III-B: FIMO output for false positive region after aligning binding site and extending to 30 nts both sides. Here only few hits are shown. The top four motif are very low p-value and very similar to the positive motif.



Motif	Sequence Name	Strand	Start	End	p-value	q-value	Matched Sequence
Unknown	chr1:1167456-1167527	−	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:6200627-6200698	−	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:6312232-6312303	−	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:11866194-11866265	−	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:12104830-12104901	−	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:17247562-17247633	−	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:27381033-27381104	−	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:30939611-30939682	−	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:31218743-31218814	+	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:33896710-33896781	+	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:41469703-41469774	+	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:42423841-42423912	−	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:44085013-44085084	−	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:63788149-63788220	+	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:111043795-111043866	−	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:113845986-113846057	−	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:204329071-204329142	−	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:204668433-204668504	−	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:206736481-206736552	+	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:206804669-206804740	+	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:222082065-222082136	+	31	41	1.38e-07	0.000404	GTCACGTGGCC
Unknown	chr1:226741737-226741808	−	31	41	1.38e-07	0.000404	GTCACGTGGCC

Fig III-C: FIMO output for positive region after aligning binding site and extending to 30 nts both sides. Here only few hits are shown.

Appendix IV

Logo plot of extended wider region 161 nts (71+11+75)

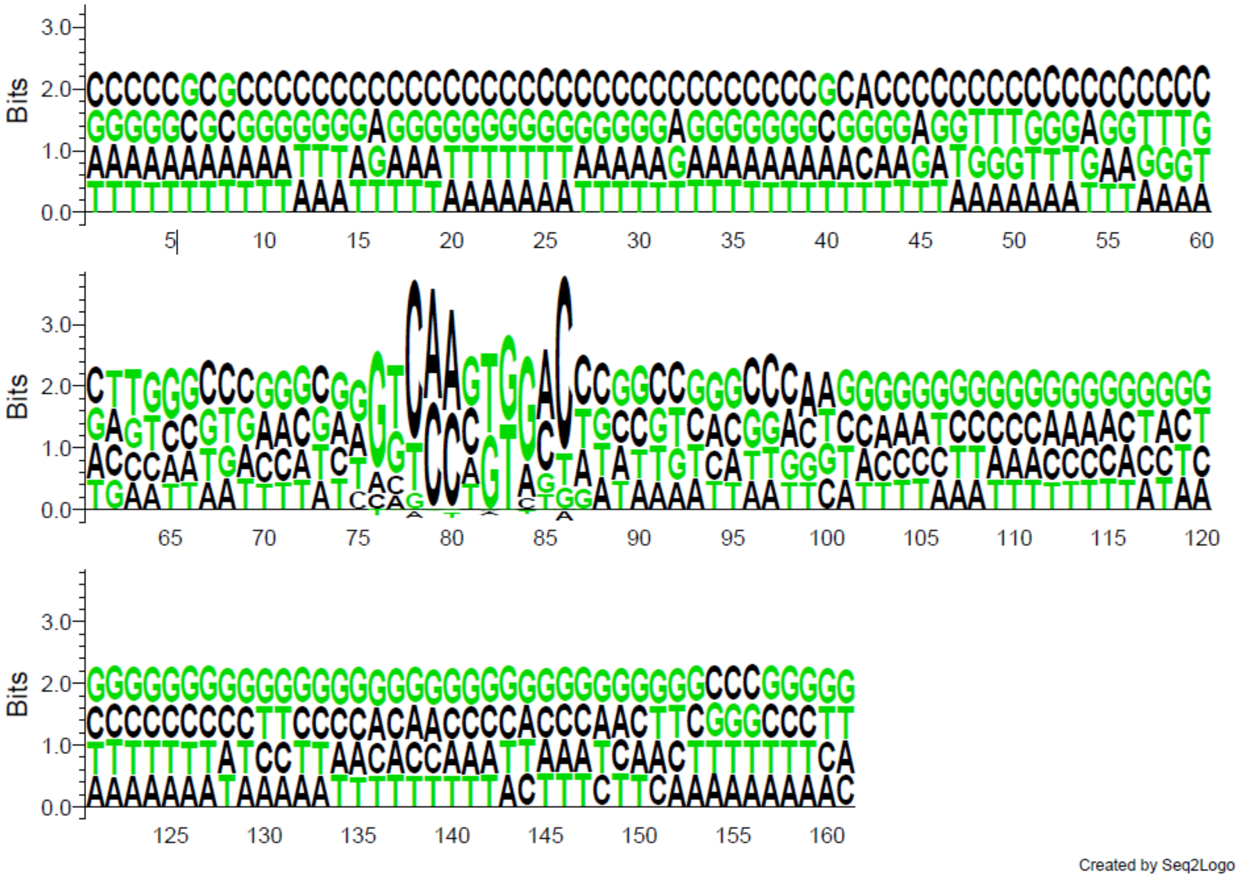


Figure IV-A: Logo plot for extended region of 161 nts

Appendix V

Description of provided motif

MEME version 4.4

ALPHABET= ACGT

strands: + -

Background letter frequencies (from uniform background):

A 0.25000 C 0.25000 G 0.25000 T 0.25000

MOTIF Unknown Unknown

letter-probability matrix: alength= 4 w= 11 nsites= 598 E= 7.7e-1029

0.143813	0.016722	0.839465	0.000000
0.008361	0.051839	0.038462	0.901338
0.000000	0.998328	0.001672	0.000000
1.000000	0.000000	0.000000	0.000000
0.000000	0.951505	0.015050	0.033445
0.239130	0.030100	0.730769	0.000000
0.010033	0.008361	0.011706	0.969900
0.000000	0.000000	0.996656	0.003344
0.341137	0.173913	0.406355	0.078595
0.080268	0.464883	0.254181	0.200669
0.060201	0.600334	0.225753	0.113712