EXPLORING TRANSCRIPTION PATTERNS AND REGULATORY MOTIFS IN ARABIDOPSIS THALIANA

by

VISHAL BAHIRWANI

B.E., Rajiv Gandhi Proudyogiki Vishwavidyalaya, India, 2006

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2010

Approved by:

Major Professor
Doina Caragea

Copyright

Abstract

Recent work has shown that bidirectional genes (genes located on opposite strands of DNA, whose transcription start sites are not more than 1000 basepairs apart) are often co-expressed and have similar biological functions. Identification of such genes can be useful in the process of constructing *gene regulatory networks*. Furthermore, analysis of the intergenic regions corresponding to bidirectional genes can help to identify regulatory elements, such as transcription factor binding sites. Approximately 2500 bidirectional gene pairs have been identified in *Arabidopsis thaliana* and the corresponding intergenic regions have been shown to be rich in regulatory elements that are essential for the initiation of transcription. Identifying such elements is especially important, as simply searching for known transcription factor binding sites in the promoter of a gene can result in many hits that are not always important for transcription initiation. Encouraged by the findings about the presence of essential regulatory elements in the intergenic regions corresponding to bidirectional genes, in this thesis, we explore a *motif-based machine learning approach* to identify intergenic regulatory elements. More precisely, we consider the problem of predicting the transcription pattern for pairs of consecutive genes in *Arabidopsis thaliana* using motifs from *AthaMap*[1] and *PLACE*[2]. We use machine learning algorithms to learn models that can predict the direction of transcription for pairs of consecutive genes. To identify the most predictive motifs and, therefore, the most significant regulatory elements, we perform *feature selection* based on mutual information and *feature abstraction* based on family or sequence similarity. Preliminary results demonstrate the feasibility of our approach.

[1] http://www.athamap.de/
[2] http://www.dna.affrc.go.jp/PLACE/

Table of Contents

List of Figures

List of Tables

Acknowledgments

My thesis not only represents the outcome of my research at Kansas State University (KSU), it also reflects the knowledge that I have gained from inspirational people that have played a major role in my progress as a student.

Dr. Doina Caragea, assistant professor with the department of *Computing and Information Sciences* at KSU and major adviser for my thesis, has taught me to perceive the goals with enough clarity that my efforts are always aligned with them. The idea of refining count and score features derived from motif characteristics, to improve the prediction of transcription patterns, is a result of many thorough discussions I had with her. I wish to acknowledge her participation in the work presented in this thesis, and thank her for showing faith in me and guiding me in times of failure and success.

Dr. Susan Brown, professor with the department of *Biology* at KSU and a member of my thesis committee, has educated me with fundamentals of *Bioinformatics*, which have been very useful in my research. I wish to thank her for her advises and for guiding me throughout the course of this research.

I am grateful to Dr. Gurdip Singh, professor, head of *Computing and Information Sciences* department at KSU and member of my thesis committee, for generously offering his time and expertise to better my work.

Furthermore, I would like to acknowledge Dr. Volker Brendel, Bergdahl Professor of Bioinformatics with the departments of Genetics, Development and Cell Biology, and Statistics at Iowa State University, for providing me with data for the experiments in this thesis and for his valuable suggestions.

I am also thankful to staff in the department of *Computing and Information Sciences* at KSU for their good natured support and for providing me with the resources needed to materialize ideas into an accomplished thesis.

At last, I wish to thank my friends and colleagues, especially Kabeer Jasuja, Swarnim Kulkarni and Sandeep Solanki for their valuable discussions, comments and suggestions.

The research described in this thesis was supported in part by a grant from the National Science Foundation (NSF 0711396) and a seed grant from the Ecological Genomics Institute at Kansas State University.

Dedication

I dedicate this thesis to my father, Dr. Ramesh Bahirwani, who has taught me that setbacks prepare us to overcome harder challenges in life.

I also dedicate it to my mother, Mrs. Vinita Bahirwani, who has taught me that hard work paired with dedication helps us achieve any goal that we wish for.

My twin, Vikas Bahirwani, has always had faith in me. This thesis would not be complete without dedicating it to him as an acknowledgment for everlasting support and affection.

Chapter 1

Introduction

Characterization of regulatory mechanisms by which plants sense and respond to *abiotic stresses* (such as drought, low temperature, high salinity) at the molecular level is crucial to understanding the responses of organisms to environmental changes. Such stresses are among the most significant factors involved in plants' adaptation to environmental changes. Identifying the genes that respond to environmental stimuli is a research problem in computational genomics and in order to understand how plants react to abiotic stress, researchers study genes and *gene regulatory networks* governing plant responses [Zhang et al., 2005].

In order to better understand gene regulation, researchers need to thoroughly identify and characterize *transcription factor binding sites* (TFBS) in genomic sequences. TFBS are sites where transcription factors, i.e., proteins that control the process of gene transcription, bind. These binding sites are largely located in the intergenic regions between genes. Analysis of the intergenic regions can lead to insights into what binding sites are important for transcription.

Towards this goal, we address the problem of predicting the transcription direction of pairs of consecutive genes over a genome, using binding sites as predictive features, especially those sites which are located in the intergenic region between two genes. More precisely, learning algorithms are provided with examples consisting of pairs of consecutive genes, represented using binding sites or, equivalently, the corresponding transcription factors. Class labels are given by the transcription direction for the two genes in the pair. While

learning to predict the transcription direction is an interesting problem in itself, it can also help us identify binding sites (a.k.a., motifs) that are important for transcription.

In this chapter, Section 1.1 describes the importance of gene regulatory networks and transcription patterns in *Arabidopsis thaliana* and Section 1.2 defines the classification problem we will address in this thesis.

1.1 Motivation

Gene regulatory networks govern functional development and biological processes of cells in all living organisms [Needham et al., 2009]. To understand the differences between cells within a species or between species or between healthy and diseased cells, researchers need to understand how genes are expressed [Davidson and Levine, 2005]. The importance of studying such networks can be gauged from the fact that discovery of complete gene regulatory networks in plants would allow the development of stress resistant crops.

Traditionally, the study of abiotic stress was carried out by perturbing optimal growth conditions and inferring a gene's function from the observed changes in gene expression. Collecting necessary data through perturbation experiments (for example, gene switches) is expensive. Furthermore, regulatory motifs are important for understanding such networks, but they are hard to find, and as a result the problem becomes more challenging. Hence, researchers are using computational methods to understand how genes are wired together to form functional networks.

With high-throughput technologies, it is now feasible to develop new and effective methods for systematic characterization of regulatory networks in plants, in response to multiple stresses. Such studies will provide insight for understanding the underlying interactions between components controlling the activities of genes involved in plants adaptation to abiotic stresses. Computational methods are now generating a plethora of putative transcription factor binding sites[1] by searching for overrepresented DNA patterns upstream of function-

[1] transcription factor binding site, motif or regulatory element are synonyms

ally related genes (i.e., genes with similar expression patterns or functional annotation). The abundance of both computationally and experimentally derived binding sites and their growing use in defining gene regulatory networks and deciphering the regulatory mechanism of individual genes make them important tools for computational biology. Discovering gene regulatory networks in *Arabidopsis thaliana* is a complex process.

Motifs have biological significance and provide strong hypotheses about the links in a regulatory network. For instance, a gene whose promoter consists of well-known regulatory elements is likely to be regulated by the transcription factors having these elements as their binding sites. Obviously, the gene being regulated and the genes encoding the transcription factors are parts of the same network. Given that regulatory elements in non-coding regions often control gene expressions and that a gene's location in a regulatory network is essential to understand its function, characterization of non-coding regions has become very important.

Computational work by [Trinklein et al., 2004] have identified approximately 2500 *bidirectional* genes, i.e., genes located on opposite strands of DNA with their transcription start sites not more than 1000 base pairs apart. Bidirectional genes are known to be co-expressed. Intergenic regions between pairs of bidirectional genes are rich in regulatory elements, which are sometimes shared by the two genes. In this thesis, we will study pairs of consecutive genes, including genes whose transcription start sites may be more than 1000 base pairs apart. We will refer to such genes as *gene pairs* (Fig. 1.1). It is important that we thoroughly study genes pairs and their corresponding intergenic regions to get a better understanding of regulatory elements.

Wang et al. [2009] have identified several thousand bidirectional gene pairs in *Arabidopsis thaliana*, with intergenic regions rich in regulatory elements. Simply searching for known transcription factor binding sites in gene promoters using motif databases such as AthaMap [Blow et al., 2009; Galuschka et al., 2007; Steffens et al., 2004, 2005] and PLACE [Higo et al., 1998, 1999] will result in many false positive motifs that are not nec-

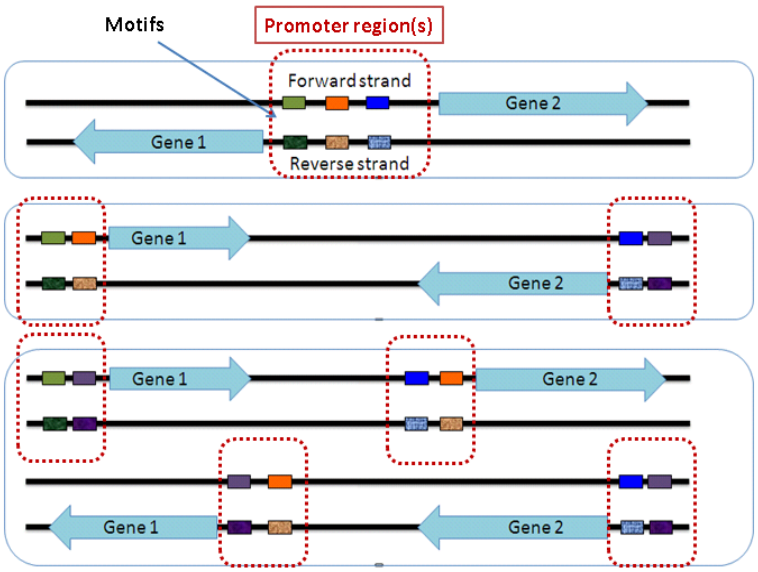

Figure 1.1: *Gene pairs and their corresponding regulatory elements.*

essarily important for initiation of transcription. A different approach is required to filter "relevant motifs" from motifs reported in these and other similar databases. Thus, the questions that we address are the following: Are all the motifs reported by databases essential for transcription? How can we identify essential motifs? Is the motif information available in current databases complete? In other words, do they contain all essential motifs?

1.2 Problem Definition and Overview of the Proposed Approach

To address the questions above, we will present *a motif-based machine learning approach* that can help to identify intergenic regulatory elements important for transcription. Specifically, we consider the problem of predicting the transcription direction for pairs of consecutive genes in *Arabidopsis thaliana* using motifs from AthaMap, PLACE and k-mers (where $k = 3, \cdots, 8$) (under the assumption that specific motifs are not available). All prediction experiments will be conducted using Weka's [Witten and Frank, 1999; Witten et al., 1999] implementations of machine learning algorithms to learn models that can predict the direc-

tion of transcription. We will also perform *feature selection* (using Weka's implementation for information gain criterion (InfoGainAttributeEval) along with Ranker's search method) and *feature abstraction* (based on sequence similarity) to identify and rank most significant (or predictive) regulatory elements.

We formulate the problems of predicting the direction of transcription for pairs of consecutive genes (Fig. 1.2) as a classification problem as follows:

Three-class problem: Given a data set $\mathcal{D} = \{((g_{i,1}, g_{i,2}), c_i)\}_{i=1,\cdots,n}$ of pairs of consecutive genes $g_{i,1}$ and $g_{i,2}$ over the alphabet Σ of nucleotides, $|\Sigma| = 4$, $g_{i,1}, g_{i,2} \in \Sigma^*$ along with their class labels c_i that belong to a finite set C, the task is to produce a model that is able to predict the class label $c \in C$ for a novel pair of consecutive genes (g_1, g_2). The class label associated with each pair of consecutive genes represents the direction of transcription for the corresponding pair: forward-reverse (FR) if the direction of transcription of g_1 is forward and of g_2 is reverse, reverse-forward (RF) if the direction of transcription of g_1 is reverse and of g_2 is forward, and forward-forward, reverse-reverse ($FFRR$) if the directions of transcription of g_1 and g_2 are either forward-forward or reverse-reverse. Our experiments will focus on this problem.

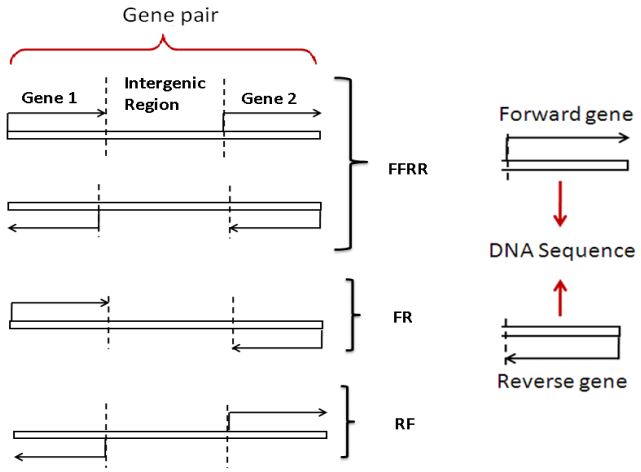

Figure 1.2: *Gene pairs and associated transcription patterns.*

Two-class problem: The problem is similar to the three-class problem, the only difference

being in the number of class labels used. We create sets of "two classes" based on the *one vs. all* rule. Thus, class labels in the two-class problem are (FR+RF, FFRR), (FR+FFRR, RF) and (RF+FFRR, FR).

The intergenic regions usually evolve faster than the genes of the genome. However, motifs found in these regions play significant role in deciding the direction of transcription of adjacent genes and are more conserved. Our work shows how to use these motifs to learn models that can predict the direction of transcription for pairs of consecutive genes and identify the most predictive motifs.

Having information about transcription factors and their binding sites, can help mark directed links from regulating genes to target genes (as nodes) in the network. However, this approach is difficult and leads to false hits. The idea of identifying transcription patterns and regulatory motifs that are predictive of such patterns can be a workaround. The latter problem is relatively simpler and helps to identify important binding sites, possibly true hits in the former (note that the latter problem is a subproblem of the bigger problem - constructing gene regulatory networks).

This thesis is organized as follows - Chapter 2 provides biological background, with focus on *motifs*, describes various biological databases storing motifs, and types of motifs that can be collected from these databases. It also describes the characteristics of motifs that can be used to derive features for machine learning algorithms; and other types of features used for learning transcription patterns. Chapter 3 presents the approaches used to group or filter motifs, specifically feature selection and feature abstraction procedures, which are used to produce motifs that are predictive for the problem stated in Section 1.2. Chapters 4 and 5 describe the experiments performed and discuss the results obtained when predicting transcription patterns. A discussion of the related work can be found in Chapter 6. Finally, we conclude our work and present several directions for future work in Chapters 7 and 8.

Chapter 2

Background

This chapter presents the background information that will be needed for understanding the work presented in this thesis. Sections 2.1, 2.2 and 2.3 describe biological background, characteristics of motifs that we exploit to construct feature vectors and types of motifs collected from multiple regulatory elements databases, respectively.

2.1 DNA, RNA and Proteins

Central Dogma of molecular biology highlights the transfer of genetic information from DNA to RNA (through *transcription*), and from RNA to protein[1] (through *translation*). DNA can be seen as a long-term copy of genetic material, while RNA is a temporary intermediary between DNA and proteins. Proteins are physical manifestations of the abstract information contained within a genome.

DNA contains genes, some of which encode for proteins. Proteins are needed to execute cell processes. The information in DNA is encoded with four chemical bases: adenine (A), cytosine (C), guanine (G) and thymine (T). The order of these bases determines the information needed for building and maintaining an organism. DNA bases pair up with each other, A with T and C with G, to form units called *base pairs* (Fig. 2.1 [NLM]). The genetic code can be seen as a set of rules by which information encoded in genetic material (DNA or RNA sequence) is translated into proteins (or amino acid sequence). Each three-nucleotide

[1]Organic compound made of amino acids sequences

combination (also known as a *codon*) designates one of the 20 amino acids (note that the code is redundant, in the sense that several codons can code for the same amino acid). An organism's DNA contains regulatory sequences and intergenic segments that contribute to phenotype, and do not get converted to amino acids during translation. Such regions, which do not carry genetic information, are involved in regulating genes.

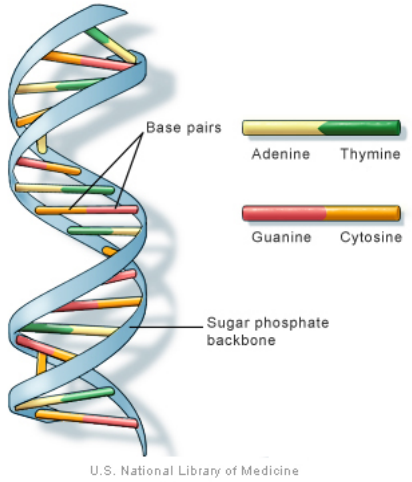

Figure 2.1: *Base pairs within DNA structure.*

The genome of an organism contains thousands of genes, but not all of these genes are active at any given moment. A gene is expressed when it is being transcribed into RNA (which is later translated into a protein). Biological mechanisms control the expression of genes, meaning that proteins are produced only when needed by the cells. Fig. 2.2 is a simplified picture of how proteins regulate genes. For transcription to occur, transcription factors need bind to regulatory regions in the promoter of the gene that they regulate. More precisely, they need to bind to regulatory elements in DNA. Once the transcription factors are bound to DNA, RNA polymerase also binds and starts transcribing the coding regions into mRNA, which is finally translated into a new protein.

Transcription factors bind themselves to the promoter of genes, either promoting (as an activator) or inhibiting (as a repressor) the transcription of genes. Hence, an associated adjacent gene is either up-regulated or down-regulated. DNA sequence that a transcription

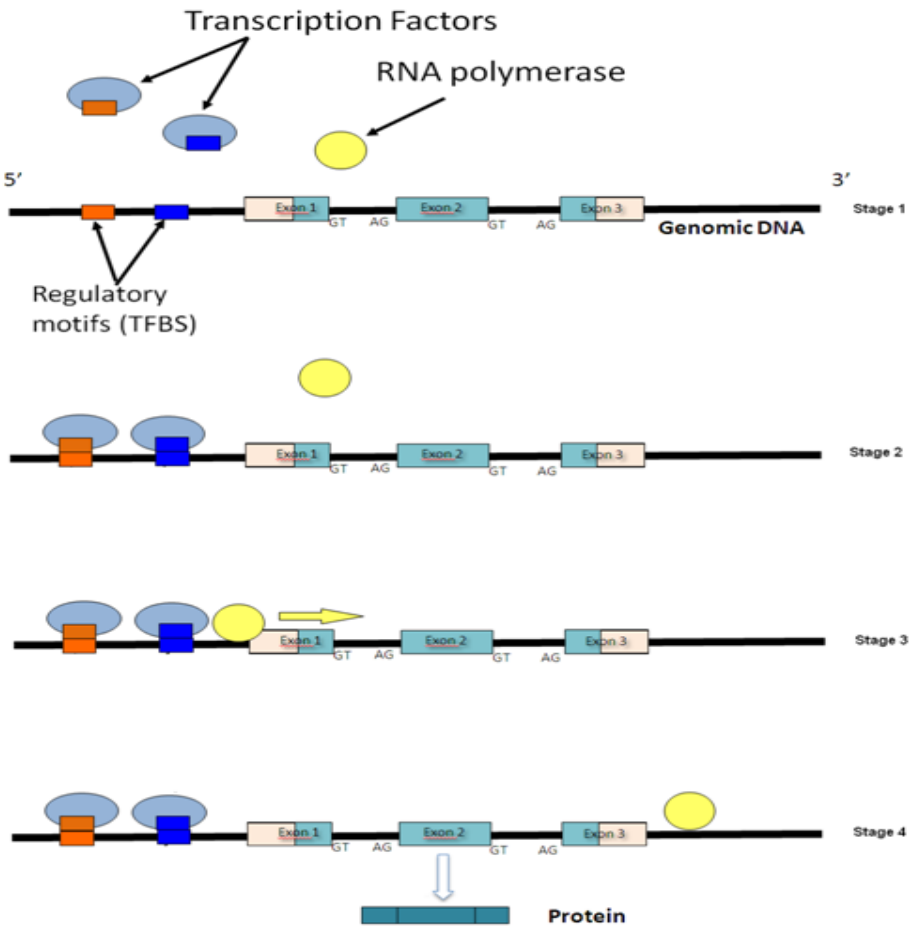

Figure 2.2: *Regulation of genes.*

factor binds to, is called as a *transcription factor binding site* or *motif*.

Having presented an overview of the central dogma of molecular biology, we define a *gene regulatory network* as a collection of genes which interact with each other (indirectly through their protein expression products), thereby governing the rates at which genes in the network are transcribed into RNA, which are further converted to proteins. Regulatory proteins are encoded by genes and therefore we have complex gene regulatory networks, including positive and negative feedback loops [Schlitt and Brazma, 2007]. Fig. 2.3 [Schlitt and Brazma, 2007] is a simple representation where genes shown encode transcription factors, that control the activity of genes encoding transcription factors.

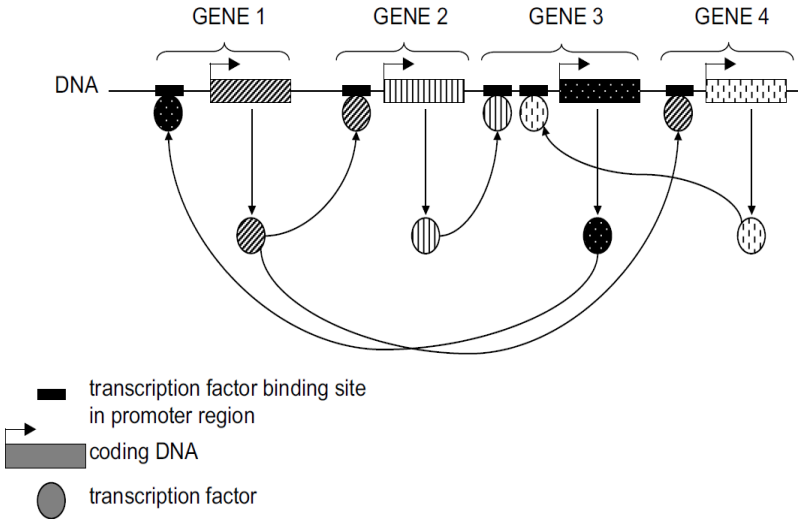

Figure 2.3: *A fictional gene regulatory network showing gene regulation mechanism.*

2.2 Motif Characteristics

As mentioned above, *motifs* are regions of DNA that play role in the regulation of gene expression [Lee and Mahato, 2009]. These elements are often the binding sites of one or more transcription factors and generally, are found in 5'-untranslated and 3'-untranslated regions of the gene of interest and, especially, in the intergenic regions. In this thesis, these regions combined are referred as *region of interest* (Fig. 2.4).

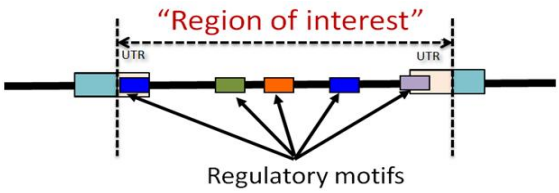

Figure 2.4: *A picture showing the "region of interest" between two consecutive genes.*

A gene pair (specifically, region of interest) can be represented using binding site information (namely the existence of the binding site in the region of interest, or equivalently *the transcription factors* corresponding to binding sites or their *families*, and *position specific scores*); and general sequence characteristics (namely *sequence length* and *GC content*). These are features that effectively incorporate the available prior knowledge needed to train machine learning classifiers. In our work, we provide classifiers with a *feature vector* of the form $\{feature_1, feature_2, feature_3 \ldots feature_k, Class\_Label\}$ and train them to predict directions of transcription (or class label) of the consecutive genes encoded in a gene pair.

Given the large amount of information that exists in biological databases, an effective and appropriate use of the data to train classifiers and to combine information from multiple sources to alleviate effects of missing or unknown information is important. We design experiments so that effective learning from aforementioned features is addressed. We collect motifs from two databases: AthaMap [Blow et al., 2009; Galuschka et al., 2007; Steffens et al., 2004, 2005] and PLACE [Higo et al., 1998, 1999]. Note that, different sets of motifs present different learning information to the classifiers with the goal of identifying the most predictive motifs.

2.3 Types of Regulatory Motifs

Pattern recognition programs used by biological databases like AthaMap and PLACE identify different types of putative transcription factor binding sites based on screening parameters supplied. Depending on whether positional weight-matrices or experimentally verified

single sites based on consensus sequences were used for screening gene sequences, output patterns can be classified as *matrix-based* or *pattern-based* motifs, respectively.

Before describing the two types of output patterns, it is worth discussing the biological databases that the motif information came from. Section 2.3.1 describes Athamap and PLACE databases, followed by Sections 2.3.2 and 2.3.3, which describe the the two types of motifs, respectively.

2.3.1 Biological Databases

This section talks about biological databases from which motifs information were collected.

A number of databases for cis-regulatory elements and gene-expression analysis that provide data for bioinformatic research are available, including several that contain information about *Arabidopsis thaliana*. We have chosen to use AthaMap and PLACE in this work, because they are in public domain, and provide online tools to search binding sites in user-selected genes (Fig. 2.5) or at specific genomic positions [Blow et al., 2009]. Database specific details are as follows:

- **AthaMap** provides a genome-wide map of potential transcription factor binding sites in *Arabidopsis*. The data in AthaMap is based on published transcription factor binding specificities available as alignment matrices or experimentally determined binding sites [Steffens et al., 2005]. Using a pattern search program called *Paster*, *matrix-based* and *pattern-based* screenings are performed to identify genomic positions of putative binding sites. A site is reported as a putative binding site by comparing its score with the threshold and maximum scores determined by Patser. There are 109 transcription factors identified by AthaMap, that are used as features in our classification experiments. Please note, AthaMap denotes motifs by the transcription factors that bind to them. Fig. 2.6 shows a partial screenshot of search results from AthaMap for gene AT1G01050 (*Arabidopsis* genome identification number). Note that, the binding site "ggaaaaagcga" and the associated transcription factor "DOF2" that binds to the

given site can be seen as two equivalent features. We will use these two features interchangeably if the transcription factor corresponding to the binding site is known, when constructing feature vectors for predicting transcription patterns. For each gene pair, we use its position coordinates to extract putative motifs spread across the region of interest. Initial experiments performed (not discussed in the thesis) indicate 0% restriction to highly conserved binding sites as the best level, while collecting motifs over a scale of $0 - 100\%$ restrictions with 10% step increase.

- **PLACE** is a database of motifs found in plants (not limited to *Arabidopsis*). These motifs have been reported in previously published papers. In addition to the motifs originally reported, their variations in other genes or in other plant species are also compiled [Higo et al., 1998]. The database reports 73 *Arabidopsis*) specific motifs to be used as features in classification experiments. A query sequence is searched for presence of these motifs using a homology search tool called as *Signal Scan* [Prestridge, 1991]. Fig. 2.7 shows an example of input file supplied to PLACE search engine, highlighting gene pair name, its transcription pattern and DNA sequence. Fig. 2.8 shows a partial screen shot of search results from PLACE for the given input file.

2.3.2 Matrix-based Motifs

Binding sites determined by matrix-based screenings fall into this category. AthaMap screens genomic sequences with the matrices (or profiles) of known transcription factors with a threshold score and reports sites with scores greater than or equal to the threshold. PSSMs predict novel binding sites solely on the basis of nucleotide frequencies at single matrix positions. Such motifs are associated with a profile, maximum and threshold scores. Fig. 2.9 shows a transcription factor ZAP1 with WRKY(Zn), 12.26 and 8.48 being its transcription family, maximum and threshold scores, respectively. AthaMap lists 51 matrix-based motifs (to be used as features in machine learning experiments). On the other hand, PLACE reports no matrix-based motifs.

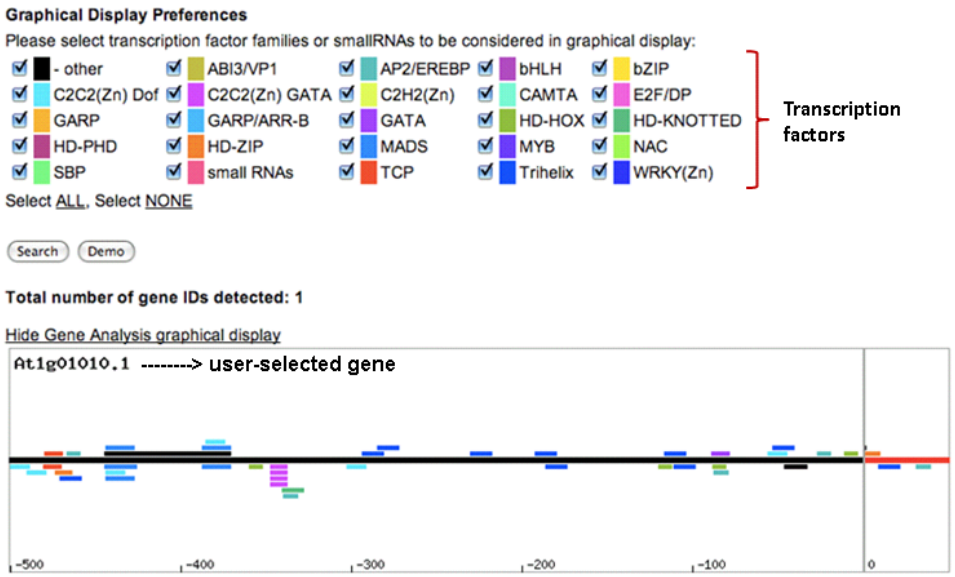

Figure 2.5: *Search results from AthaMap showing transcription factors that bind to a gene.*

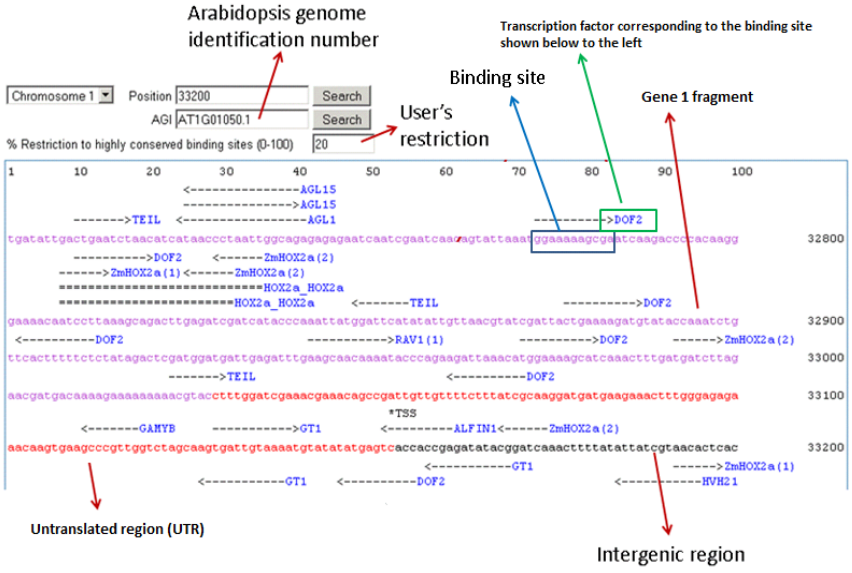

Figure 2.6: *A screen shot of AthaMap's output for gene AT1G01050 at 20% restriction level.*

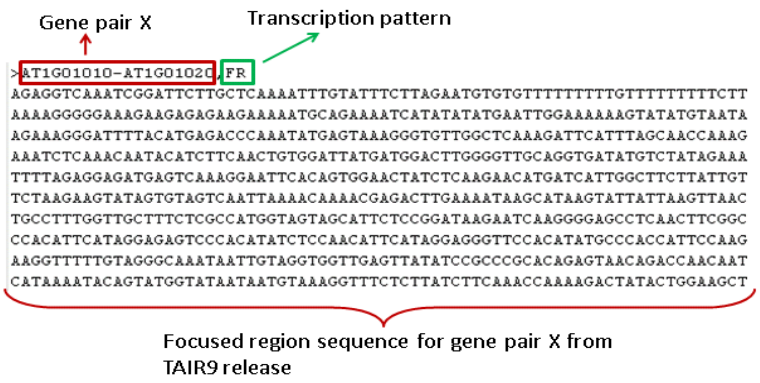

Figure 2.7: *A sample input given to PLACE in FASTA format.*

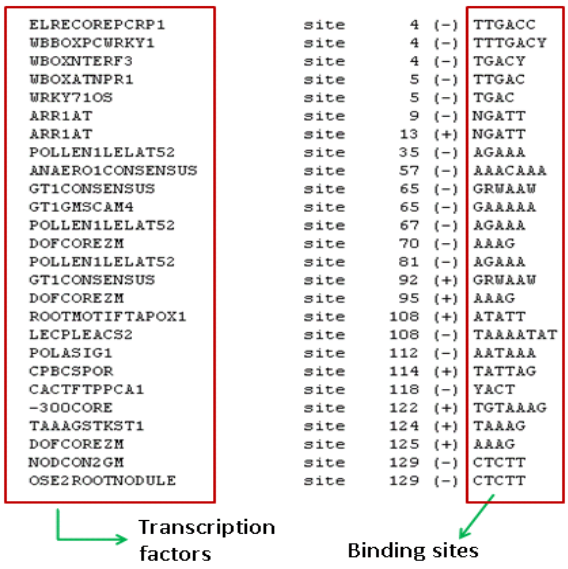

Figure 2.8: *A screen shot of search results from PLACE for gene pair AT1G01010-AT1G01020.*

Name: ZAP1
Family: WRKY(Zn)

Matrix:

A |	0	0	0	45	0	0	1	33	2
C |	0	0	0	0	45	44	3	6	4
G |	0	0	45	0	0	0	39	3	35
T |	45	45	0	0	0	1	2	3	4

Max. score: 12.26

Threshold: 8.48

Figure 2.9: *An example of a matrix-based motif ZAP1.*

2.3.3 Pattern-based Motifs

Binding sites determined by pattern-based screenings fall into this category. AthaMap and PLACE screen genomic sequences with binding sequences (these are experimentally verified) gathered from known transcription factors. Each pattern-based motif is associated with a factor and a family name, a verified binding sequences where the factor binds and corresponding consensus pattern (Fig. 2.10). AthaMap lists 58 and PLACE lists 73 pattern-based motifs, respectively.

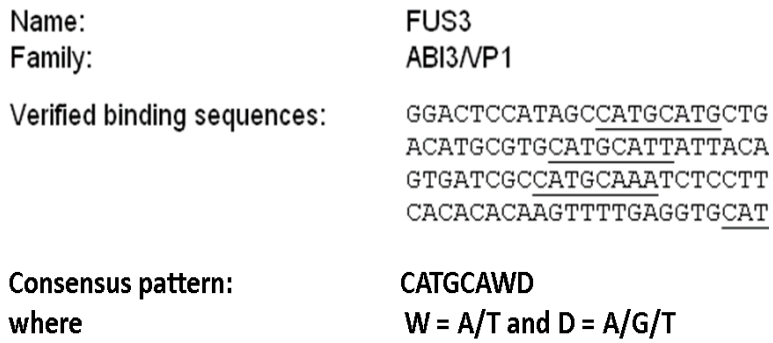

Figure 2.10: *An example of a pattern-based motif FUS3.*

2.4 Feature Representation

Each example in our data set $\mathcal{D}$ is represented as "100*bp of* $gene_1$*, intergenic region,* 100*bp of* $gene_2$" - together forming the region of interest. We first collected motifs from AthaMap and PLACE. AthaMap identifies 109 and PLACE identifies 73 motifs for *Arabidopsis thaliana* genome (consisting of 30270 gene pairs from all five chromosomes). We then encoded each gene pair using the *bag of motifs* representation [McCallum and Nigam, 1998]; i.e. for both AthaMap and PLACE we construct separate data sets where each instance is a vector of 109 and 73 features, respectively plus one class label. Each position in the vectors is a feature that represents the number of times the corresponding motif appears in a given example. By feature (Fig. 2.11) we mean count or score of a characteristic that can be used to learn statistical models and predict unknown properties. Based on how we deal

with these features, there are two types of feature representations and each representation has significant effect on performance of classifiers (Fig. 2.12). The next subsection provides more insights into two main feature representations.

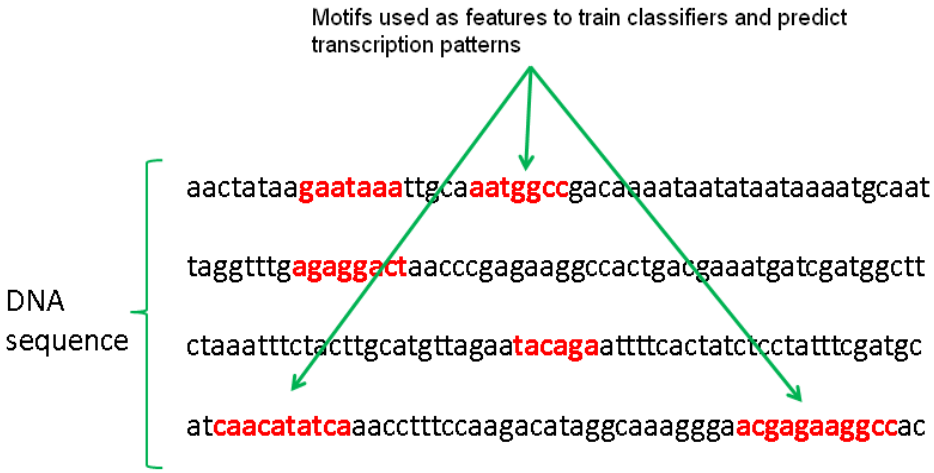

Figure 2.11: *Binding sites found in DNA sequence.*

2.4.1 Count Representation

When we count the number of times each motif appears in a given gene pair sequence, we represent it as a feature using the *count*. We may consider locations of motifs in the region of interest. To capture this, we record the presence or absence of a motif in gene1, intergenic or gene2 regions. Count representation works with both matrix-based and pattern-based motifs. We will conduct experiments with various permutations and combinations on the data (AthaMap or PLACE), types of features (matrix-based or pattern-based) and types of feature representations (count or score) to find efficient ways of training classifiers and predicting transcription patterns.

2.4.2 Score Representation

If a motif is identified by matrix-based screening, then it has a score associated with it. We can use this score as a feature. For each motif that is reported in a sequence, we take

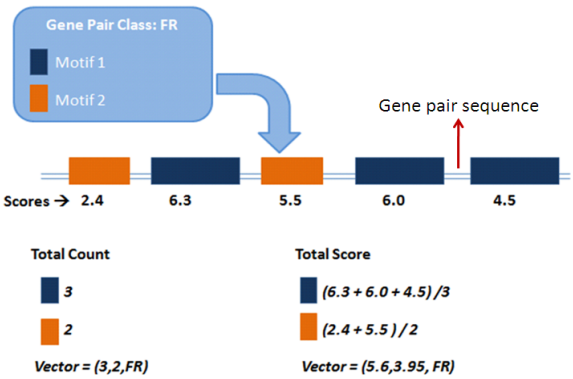

Figure 2.12: *An example of feature representations.*

an average of its scores at various occurrences and use the same in place of total count (Fig. 2.12). This approach is not available with pattern-based motifs since no scores are associated with them.

Chapter 3

Feature Selection and Abstraction

A prediction problem is defined as the task of inferring a function from training examples of its input and output [Mitchell, 1997]. The function to be learned is called target concept (denoted by c), and the set of items over which it is defined is called the set of instances (denoted by X), i.e. $c : X \rightarrow \{class_1, class_2, class_3 ... class_k\}$. For any instance $x \epsilon X$, the value $c(x)$ is called the class label for x [Mitchell, 1997]. For the three-class problem that we consider, $c(x)$ is a function of the form $c : X \rightarrow \{FFRR, FR, RF\}$. When learning a target concept, the learner is presented with a set of *training examples*, each consisting of an instance x from X, along with its value $c(x)$. The output of the learner is an estimate of the function c (called a classifier), which can predict the class label for new unlabeled instances (called test instances) [Mitchell, 1997].

Each prediction problem is associated with a performance measure P, which is used to evaluate the learning process. The main goal of our work is to improve the performance of learning algorithms at the task of predicting transcription patterns.

3.1 Types of Feature Vectors

Several types of feature vectors, described below, can be derived from sequence data and can be considered to address the prediction problems at hand. Among such vectors, we consider the following:

- Binding sites: $\{site_1, site_2, site_3 ... site_k\}$ or, equivalently, their corresponding transcription factors: $\{tf_1, tf_2, tf_3 ... tf_k\}$.
- Transcription factor families: $\{family_1, family_2, family_3 ... family_k\}$, which correspond to groupings of motifs (or transcription factors) into families.
- Motif (k-mer) clusters: $\{cluster_1, cluster_2, cluster_3 ... cluster_k\}$, when k-mers obtained using a sliding window are grouped based upon sequence similarity using a hierarchical agglomerative clustering (with the goal of reconstructing motif families).
- Gene pair sequence length: $\{GenePairLength\}$
- Region specific sequence lengths: $\{gene_1, intergenic, gene_2\}$
- GC content: $\{GC\_Content\}$

To evaluate the predictive power of these feature vectors, we will use them separately and in combinations (e.g., binding sites together with sequence length, or binding sites together with GC content) to learn classifiers. In principle, each of the feature vectors could provide the classifiers with complementary information, and our goal is to find out what features or combinations of features are the most predictive. Furthermore, we want to investigate the effect of feature selection and feature abstraction (dimensionality reduction methods) at identifying the most predictive features.

In what follows, we describe the feature selection and feature abstraction approaches that we will use to filter out noisy or irrelevant motifs from the feature set and to group similar features (e.g., motifs) into clusters of more general features (which might capture better the class information).

3.2 Feature Selection

Witten and Frank [1999] describe *feature selection* as a method used in machine learning, for selecting a subset of relevant features in order to generate more efficient learning models.

There are often two or more features that are similar to each other, and thus are not providing significantly more information than any of them individually. The idea is to select features that have high interdependence between feature values and classes. However, the interdependence among the selected features should be minimized, so that redundancy is minimized.

The feature selection criterion that we use in our experiments is based on the *information gain* criterion (or maximizing the mutual information with the class variable) [McCallum and Nigam, 1998]. We use Weka's implementation for information gain (InfoGainAttributeEval along with Ranker's search algorithm) to rank a set of features in the decreasing order of their mutual information with the class variable and we select only features for which the mutual information is above a predefined threshold.

The advantages of performing feature selection include dimensionality reduction, fast learning process, enhanced generalization capability, and better model interpretability.

3.3 Feature Abstraction

Clustering can help to find concepts in the data by providing mechanisms for grouping similar entities. Moreover, algorithms such as hierarchical clustering organize data entities in a form of hierarchy of concept-clusters providing the ability to explore derived concepts at various levels of abstraction.

Clustering is an unsupervised learning task that can be defined as the process of partitioning data into groups of similar entities, where each group corresponds to a concept [Berkhin, 2002]. Clustering algorithms are highly dependent on the selection of a distance metric that assigns a score to every pair of entities that may be grouped together. The distance metric captures the extent of similarity (or dissimilarity) between candidate pairs. In this work, we compute the distance between two clusters as the average distance among all distances between the possible pairs of entities contained in the two clusters. Hence, the clustering algorithm used is an *average-linkage* (or group-average) clustering [Manning et al., 2008,

Chapter 17].

Hierarchical clustering algorithms build a tree of clusters by successively grouping the closest cluster pairs, until no further grouping is possible. In the resulting tree (often called *dendrogram*), each cluster node at an intermediate level is associated with a parent cluster, one or more child nodes and one or more sibling nodes. This approach allows exploration of the data at different levels of granularity. Thus, parent nodes represent abstract notions of the concepts that their children embody [Berkhin, 2002; Manning et al., 2008]. The agglomerative approach starts by considering each instance as a distinct singleton cluster, and based on the similarity criterion, successively merges clusters together until the termination conditions are satisfied [Jain et al., 1999]. Fig. 3.1 show the result of an agglomerative algorithm which begins with singleton clusters "A", "B" and "C", and builds the hierarchy in a bottom-up fashion. We use CLUTO to group k-mers into concept-clusters.

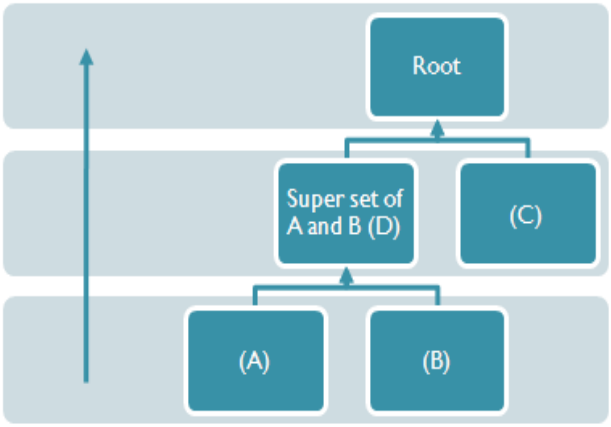

Figure 3.1: *Hierarchical agglomerative clustering.*

3.3.1 Clustering Motifs from Biological Databases

AthaMap and PLACE identify motifs and transcription factors that bind to them. Thus, there is a correspondence between motifs and transcription factors. As transcription factors can be grouped in families, we can also group motifs into families. We represent sequences as motifs both at *transcription factor level* (the motifs themselves) and at *transcription family*

level (clusters of motifs that belong to the same family).

1. AthaMap:

   For user-specified genes or genomic positions, AthaMap searches for 51 matrix-based and 58 pattern-based motifs. A simple feature vector in this case consists of 109 features (or motifs) at transcription factor level. However, each motif belongs to a family. Matrix-based motifs fall into 21 families and pattern-based motifs fall into 15 transcription families. When grouped together, we get 24 unique transcription families. Hence, the same sequence can be represented using 109 motifs at the transcription factor level, or 24 abstract motifs at the transcription family level.

2. PLACE:

   For user-specified genes, PLACE searches for 73 pattern-based motifs. We grouped them into 48 abstract features, based upon respective transcription families and binding sequence similarity.

Factor and family level motifs can be arranged in a hierarchy, as shown in Fig. 3.2. Here, motifs ARF1 and AtLEC2 are grouped in the ABI3 family. We will train classifiers at both levels of abstraction and analyze performance of the learned models.

3.3.2 Filtering and Clustering *k-mers* (Unbiased Approach)

While many motif databases are available for *Arabidopsis*, this may not be the case for other less studied organisms. Therefore, we switch our attention to information that is unknown and yet unavailable in databases. The approach of collecting motifs directly from gene pairs' DNA sequences is named the *unbiased approach* and motifs collected in the process are termed as *k-mers*, where k is length of a motif. We want to investigate the ability of learning algorithms to make use of k-mers in cases where not many known motifs are available. To identify potentially useful k-mers, we use observations that we made based on the information found in databases.

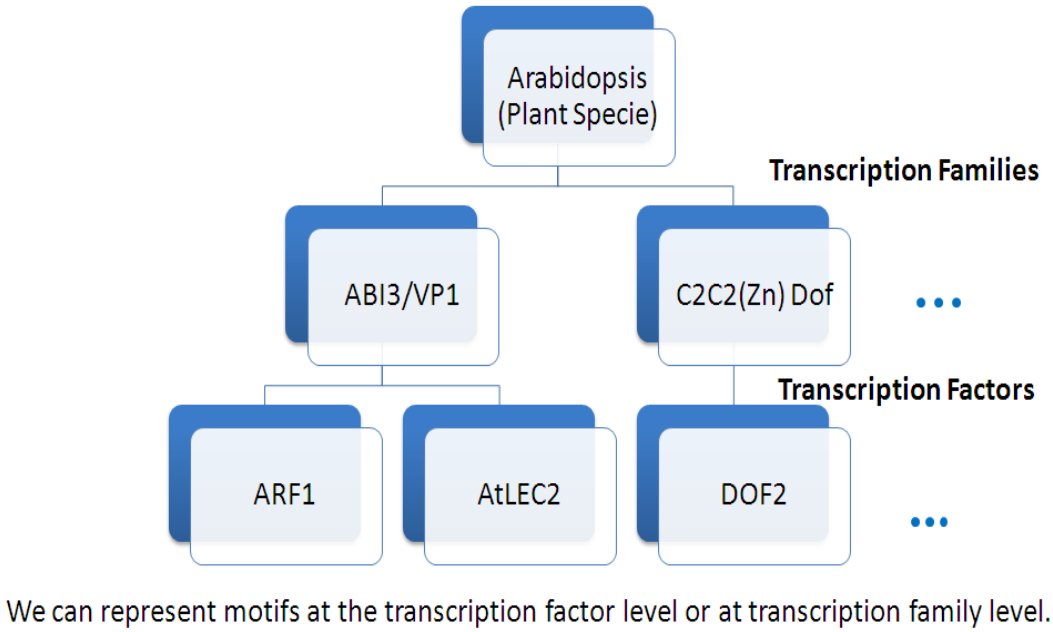

Figure 3.2: *Hierarchical organization of motifs collected from AthaMap.*

First, motifs collected from AthaMap and PLACE, are of variable lengths between 4-10 basepairs. In the unbiased approach, we generate features by enumerating k-mers of variable length k- ($k = 3, \cdots, 8$) using a window-based approach. This will ensure that all important but unknown motifs will be included (while irrelevant motifs will be filtered out using feature selection).

Fig. 3.3 shows the way we collect k-mers from gene sequences. For instance, to collect all possible 5-mers, we scan a window of size equal to 5 basepairs over region of interest. From this list, we extract unique motifs to form a feature vector of 5-mers. Table 4.2 show the number of motifs collected for all k-mers that we consider in this work.

As can be seen in Table 4.2, a large number of motifs are obtained for our data. Training classifiers from examples consisting of a large number of features (Table 4.2) will result in overfitting and, hence, poor performance. Therefore, we need to perform feature selection in order to filter top-ranked motifs of variable lengths. Please note, when k-mers are treated as separate feature sets, we call them as "separate k-mers" and when we combine all possible

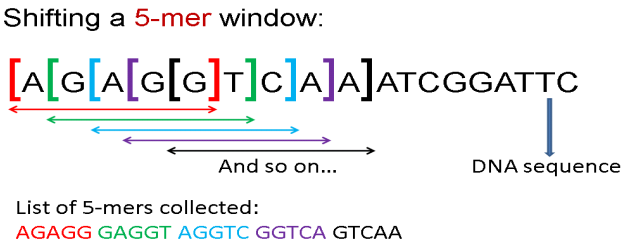

Figure 3.3: *Shifting a window of size k=5 over DNA sequence to collect all possible 5-mers.*

Table 3.1: *Motifs collected for each type of k-mers.*

k-mers	Number of Motifs
3-mer	64
4-mer	256
5-mer	1024
6-mer	4096
7-mer	16384
8-mer	65536
Total k-mers	**87360**

motifs of length 3-8 basepairs, we call them as "grouped k-mers".

Furthermore, based on previous family and factor level concepts, grouping motifs into more abstract features will result in better classifiers (this is discussed in Chapter 5). Yet, we cannot group k-mers based upon their families because the information is unknown. It is logical to cluster them based upon their sequence similarities.

1. Separate k-mers:

   In each feature set, motifs are of same length; we used *hamming distance* [HAMMING, 1950] as the measure of similarity (or dissimilarity) between two motifs, for grouping purposes.

2. Grouped k-mers:

   In this case, motifs are of variable length; we used the *end-gap free alignment tool* from EMBOSS [Sarachu and Colet, 2005], to get similarity scores for grouping purposes. Distance matrices consisting of the above calculated scores were provided to the clustering software CLUTO [Karypis, 2003].

In Fig. 3.4, CLUTO clusters 4 motifs, we get 4 feature vectors (one for each cut). We train classifiers on each cut, to find the best performing cut.

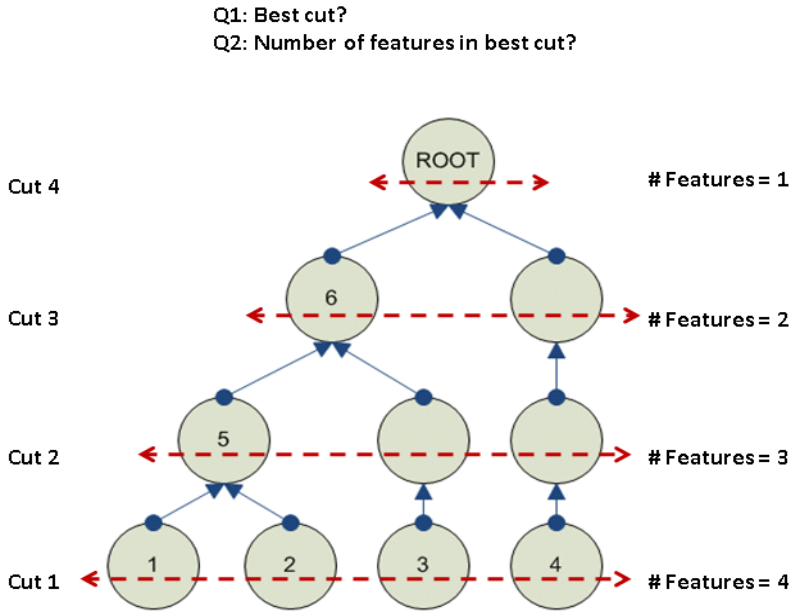

Figure 3.4: *Levels of abstraction and features at various cuts.*

Chapter 4

Experimental Setup

This chapter describes the experiments conducted to evaluate different types of feature vectors discussed in Section 3.1. We have conducted a series of experiments designed to investigate the performance of several classification algorithms at predicting transcription patterns for pairs of consecutive genes, when presented with different types of feature vectors, generated from the DNA sequence.

In each experiment, we consider the following classifiers (with default parameters), whose implementations are provided by the WEKA data mining software [Witten et al., 1999].

- Support Vector Machines (SVM) with *build logistic model* option enabled,
- Random Forests
- Logistic Regression (Logistic)

We have performed experiments on *Arabidopsis thaliana* data. Gene pairs from *Arabidopsis* genome were used to construct training and test data sets for classifiers. Data statistics are shown in Table 4.1. The data is balanced in terms of number of instances for each transcription pattern.

The performance of each algorithm is measured by the area under the *Receiver Operating Characteristic* (ROC) curve [Fawcett, 2005], i.e. the curve depicting the tradeoff between the *true positive rate* vs. *false positive rate* (Figure 4.1). The area under the ROC curve, or

Table 4.1: *Data statistics for Arabidopsis genome with* 5 *chromosomes.*

Class Label	Number of Instances
FFRR	15955
FR	7163
RF	7152
Total Instances	**30270**

AUC (shaded region in Figure 4.1), is reported on a scale from 0 to 1, 0 being the minimum value and 1 being the maximum value. Thus, higher values of the AUC indicate better performances of a classifier at a given prediction task, while lower values indicate otherwise.

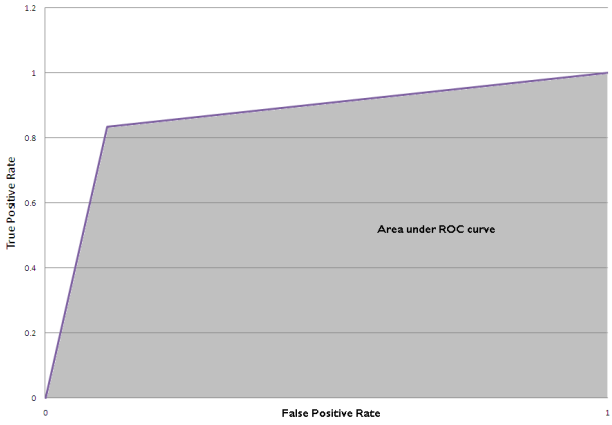

Figure 4.1: *A sample ROC curve*

The experiments in this thesis are designed to address several questions such as: What motif representation gives better results, a count representation or a score representation? Do the motifs collected from existing databases give better results than the motifs obtained by enumerating all k-mers? Do feature selection and feature abstraction approaches improve the performance? Which one is more effective? More precisely, the following experiments are performed:

1. AthaMap factor level motifs:

   For Experiment 1, we learn classifiers on 109 AthaMap motifs at factor level. Attribute values in the feature vector refer to count representation of respective motifs. This is

the simplest feature vector, that is populated with motifs from one of the databases. Initially, we do not perform feature selection and abstraction, so the AUC values for this experiment represents our baseline for motifs derived from AthaMap.

2. AthaMap family level motifs (feature abstraction):

   For Experiment 2, we learn classifiers on the 24 AthaMap motifs at family level (the goal is to capture more general motifs). Attribute values in the feature vector refer to count representation of the respective motif families.

3. PLACE factor level motifs:

   Experiment 3 is similar to Experiment 1, but with motifs from PLACE. We learn classifiers on 73 PLACE motifs at factor level. Attribute values in the feature vector refer to count representation of respective motifs. Note that, PLACE does not provide scores for motifs. The AUC values collected will be the baseline for motifs from PLACE.

4. PLACE family level motifs (feature abstraction):

   Experiment 4 is similar to Experiment 2, we generate an abstract set of features (as compared to the vector with 73 attributes), by grouping 73 *Arabidopsis* specific motifs reported by PLACE into 48 transcription families.

5. "AthaMap + PLACE" factor level motifs:

   For Experiment 5, we combine all factor level motifs from AthaMap and PLACE, to alleviate the effect of missing information from both the databases. We will combine 109 AthaMap and 469 PLACE (all plant motifs) motifs and perform feature selection to remove redundant features. AUC values hence produced, are expected to be comparable and possibly better than the baseline values.

6. Count vs. Score representations with AthaMap matrix-based motifs (factor level):

   AthaMap provides two types of motifs namely, matrix-based and pattern-based. For

the matrix-based motifs, we can consider count or score representations when creating feature vectors. For Experiments 6 and 7, we are interested to study which of these representations is more likely to result in better classifier performance while taking into account motifs at factor and family levels, respectively.

7. Count vs. Score representations with AthaMap matrix-based motifs (family level):
   Experiment 7 is similar to Experiment 6, the difference being, classifiers will be trained over 21 AthaMap matrix-based motifs at family level.

8. Feature selection over AthaMap factor level motifs:
   To improve the performance of classifiers over motifs collected from AthaMap, for Experiment 8, we use feature selection to filter the irrelevant features from the complete feature set. Based on mutual information, we generate a list of motifs in the decreasing order of information gain. Then, we select subsets of top-ranked motifs in an incremental way such as top 10, 20, 30,... and so on; to study the correlation between number of features and the AUC value for Random Forest and SVM classifiers.

9. Feature selection over PLACE factor level motifs:
   Experiment 9 is similar to the Experiment 8, except that it considers PLACE motifs.

10. AthaMap family level motifs, plus GC content as an extra feature:
    For Experiment 10 (as well as Experiments 11 and 12), we combine motifs with features obtained from the gene pair sequences, to study their contribution of the latter on performance. Hence, in the current experiment, both AthaMap motifs at family level and the GC content score of the consecutive gene pair sequence will be used to create the feature vector.

11. AthaMap factor level motifs and "gene1-intergenic-gene2" lengths as added features:
    For Experiment 11, we include both, AthaMap motifs at factor level and lengths of gene1, intergenic and gene2 sequences, that combine to form region of interest, when creating feature vectors.

12. AthaMap family level motifs and "gene1-intergenic-gene2" lengths as added features:
    Experiment 12 is similar to Experiment 11, except that the motifs used are at family level as opposed to factor level.

13. Gene pair vs. region specific motifs (AthaMap):
    In previous experiments, while collecting motifs for gene pairs we did not take into account their locations, i.e. whether motifs occurred in gene promoters or in intergenic region. For Experiment 13, we study effects of motif locations on classifiers performance. When ignoring various genic regions, feature vector lengths for factor and family level motifs are 109 and 24, respectively. While, when taking care of motif locations, feature vector lengths with factor and family level motifs are tripled (a vector for each of the three regions) i.e. $109 * 3$ and $24 * 3$, respectively. Note that, gene1, intergenic and gene2 sequences, combine to form region of interest.

    Experiments 14-17 are related to the unbiased approach (introduced in Chapter 3).

14. Learning from separate k-mers:
    For Experiment 14, we collect k-mers of length 3-8 basepairs and use them to construct feature vectors. This task is similar to Experiments 1 and 3. Classifiers are trained over different types of k-mers with number of features in each row being equal to those mentioned in the Table 4.2.

Table 4.2: *Number of features collected for each type of k-mers.*

k-mers	Number of Motifs
3-mer	64
4-mer	256
5-mer	1024
6-mer	4096
7-mer	16384
8-mer	65536
Total separate k-mers	**87360**

15. Learning from top-ranked separate k-mers (from feature selection):

To improve results from previous experiment, in Experiment 15, we do feature selection on k-mers and discard motifs with information gain score equal to zero. Table 4.2 shows basic sets of separate k-mers, while Table 4.3 shows top-ranked k-mers filtered using feature selection. This task is similar to Experiments 8 and 9.

Table 4.3: *Top-ranked motifs (mutual information* > 0.00*) in separate k-mers.*

k-mers	Number of Top-ranked Motifs
3-mer	64
4-mer	256
5-mer	540
6-mer	230
7-mer	160
8-mer	100
Total grouped k-mers	**1350**

16. Learning from top-ranked grouped k-mers (feature selection):

    For Experiment 16, we report performance of classifiers trained over 1350 grouped k-mers (Table 4.3). Here, we have included all important k-mers in the feature set.

17. Learning from "best cut" clusters of k-mers (feature abstraction over selected motifs):

    For Experiment 17, we are interested to group 1350 top-ranked grouped k-mers into concept-clusters, allowing classifiers to learn from more abstract features. We perform feature abstraction over variable length k-mers as discussed in Chapter 3.

    Since, k-mers are of variable length, they are group based on similarity scores from an end-gap free alignment tool. In turn, CLUTO clusters them, and outputs 1350 cuts along the agglomerative hierarchy. The best cut, will provide appropriate number of clusters required to train classifiers in minimum time and achieve high performance.

Chapter 5

Results

In this chapter, we will show the results of the experiments described in Chapter 4. The chapter is organized in two Sections: Section 5.1 presents the results of the experiments conducted with information collected from biological databases, and Section 5.2 presents the results of the experiments conducted as part of the unbiased approach. Section 5.3 presents results from experiments based on a new approach called *Classifier Confidence Approach*, that we experimented with based on our knowledge gained from the unbiased approach.

The AUC values that are highlighted in the tables in this chapter show the performance results of classifiers that were *statistically significantly* better than their respective baselines. Each table presents the AUC values for classifiers when predicting FFRR, FR and RF class labels (please refer to Section 1.2 for reviewing our Problem Definition). In addition, it also presents the overall performance of the classifier calculated as the weighted average of its performances for each class label.

5.1 Biological Databases Approach

In what follows, we present results from experiments 1 to 13 (Chapter 4) conducted with features derived from information available in biological databases.

1. AthaMap factor level motifs:

Results from this experiment (Table 5.1) show that Simple Logistic classifier has the best performance in predicting FFRR, FR and RF class labels. The overall performance of the classifier is 77%.

Table 5.1: *Cross-validation results with AthaMap motifs (factor level) using count representation*

Classifiers Learned	FFRR	FR	RF	Weighted Average
Random Forest	0.685	0.9	0.724	0.745
SVM - PolyKernel	0.513	0.76	0.691	0.614
Simple Logistic	**0.703**	**0.906**	**0.776**	**0.769**

2. AthaMap family level motifs:

   Results from this experiments are shown in Table 5.2. As can be seen from the table, the Random Forest classifier has the best performance; 85%, 96% and 86% in predicting FFRR, FR and RF class labels, respectively. Besides, the overall performance of the classifier is 88%.

Table 5.2: *Cross-validation results with AthaMap motifs (family level) using count representation*

Classifiers Learned	FFRR	FR	RF	Weighted Average
Random Forest	**0.849**	**0.961**	**0.864**	**0.879**
SVM - PolyKernel	0.709	0.918	0.784	0.776
Simple Logistic	0.729	0.931	0.787	0.791

   By analyzing the AUC values in Tables 5.1 and 5.2, we see that the results of the classifiers trained on family level motifs are better than those of the classifiers trained on factor level motifs. Specifically, there is a significant increase in the performance (approximately 10%) because when we move up in the motif hierarchy the feature vectors capture more general information ("semantically equivalent" motifs).

3. PLACE factor level motifs:

   Table 5.3 presents the results from this experiment. As shown in the table, Simple

Logistic classifier performs the best in predicting FFRR, FR and RF class labels with an overall performance 69%

Table 5.3: *Cross-validation results with PLACE motifs (factor level) using count representation*

Classifiers Learned	FFRR	FR	RF	Weighted Average
Random Forest	0.587	0.746	0.654	0.642
SVM - PolyKernel	0.504	0.715	0.686	0.598
Simple Logistic	**0.62**	**0.802**	**0.746**	**0.694**

4. PLACE family level motifs:

   To study the effect of transition to family level features, we grouped 73 *Arabidopsis* motifs into 48 transcription families. The results presented in Table 5.4 show that Simple Logistic classifier has the best performance; 74%, 92% and 80% in predicting FFRR, FR and RF class labels, respectively. For this experiment, the overall performance of the classifier is 80%.

Table 5.4: *Cross-validation results with PLACE motifs (family level) using count representation*

Classifiers Learned	FFRR	FR	RF	Weighted Average
Random Forest	0.671	0.877	0.715	0.731
SVM - PolyKernel	0.74	0.922	0.804	0.799
Simple Logistic	**0.745**	**0.922**	**0.805**	**0.802**

Based on the AUC values in Tables 5.3 and 5.4, we can claim that even for PLACE database, classifiers trained at family level have better performance than the classifiers trained at factor level. There is 10-12% increase in the performance of family level classifiers as compared to their factor level counter parts. Simple Logistic is the best classifier in both the cases.

Comparing the AUC values in Tables 5.1 and 5.3, we see that for Random Forest classifier, AthaMap motifs generate better feature vectors as compared to PLACE, with 10-15% rise in model's performance. Similarly, there is 10% performance difference

between Simple Logistic classifier trained over AthaMap and PLACE. However, SVM performance is similar in both cases.

When we compare AUC values in Tables 5.2 and 5.4, for Random Forest, for each class label, AthaMap features show 10-13% better performance than PLACE. However, SVM and Simple Logistic perform 2-5% better with PLACE. Overall, AthaMap is better-suited for our problem relative to PLACE.

5. "AthaMap + PLACE" factor level motifs:

   For this experiment, we grouped 109 AthaMap and 469 PLACE motifs to get a feature vector of 578 factor level motifs. Its results are shown in Table 5.5. Comparing Tables 5.1, 5.3 and 5.5, it is prominent that classifiers perform significantly better (5-10% increase) when provided with features from multiple data sources.

   Also, feature selection results in classifiers with similar performance but less training time. Fig. 5.1 shows the dependence of the AUC values on the number of features selected. The peaks of the graphs highlight the best performance. As can be seen, less than 100 best features results in best performance. As we increase the number of features, the performance decreases.

Table 5.5: *Cross-validation results with AthaMap and PLACE motifs (factor level) using count representation*

Classifiers Learned	FFRR	FR	RF	Weighted Average
Random Forest	0.644	0.841	0.692	0.703
SVM - PolyKernel	0.687	0.894	0.752	0.752
Simple Logistic	**0.75**	**0.925**	**0.81**	**0.806**

6. Count vs. Score representations with AthaMap matrix-based motifs (factor level):

   Tables 5.6 and 5.7 report performance of classifiers trained on 51 AthaMap matrix-based motifs at factor level, pertaining to count and score representations, respectively. A closer look at values in these tables identify count representation to be superior.

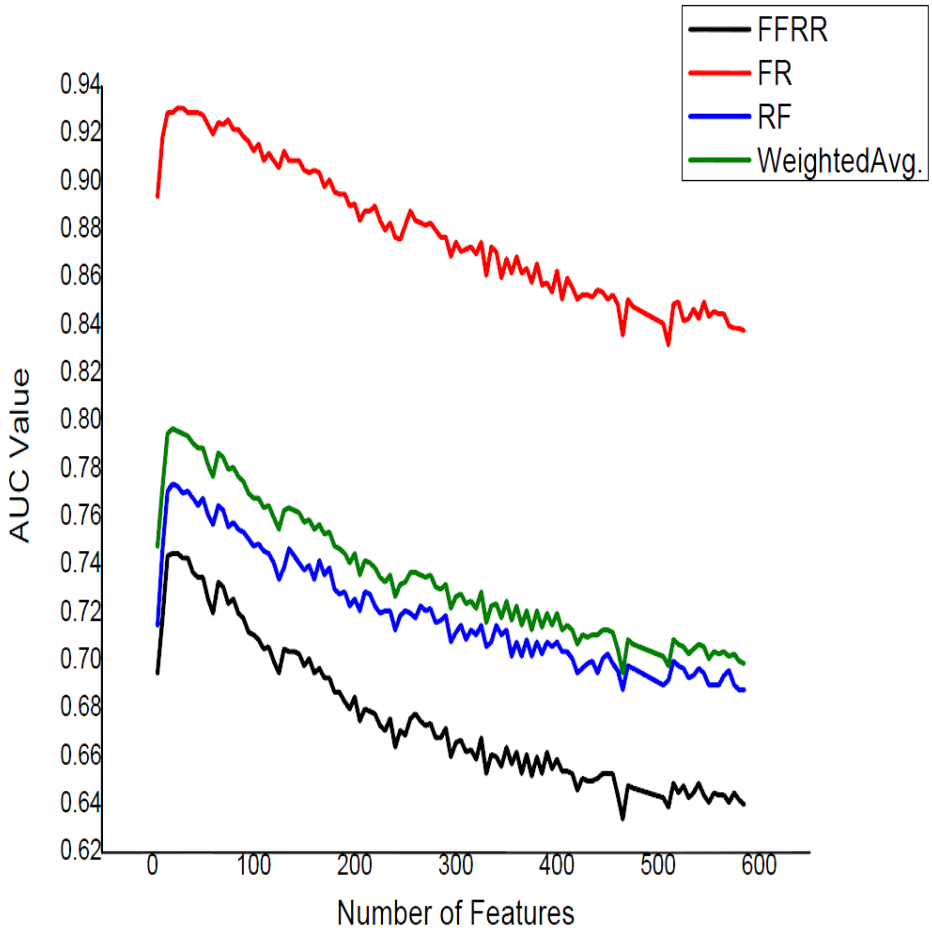

Figure 5.1: *The area under the ROC Curve as a function of number of features selected using AthaMap and PLACE motifs combined. Using a relatively small number of features (motifs), the classifiers achieve highest performance. As we add more and more features, the performance of classifiers decreases significantly.*

Table 5.6: *Cross-validation results with AthaMap matrix-based motifs (factor level) using count representation*

Classifiers Learned	FFRR	FR	RF	Weighted Average
Random Forest	0.714	0.935	0.751	0.775
SVM - PolyKernel	0.709	0.915	0.789	0.777
Simple Logistic	**0.741**	**0.938**	**0.794**	**0.8**

Table 5.7: *Cross-validation results with AthaMap matrix-based motifs (factor level) using score representation*

Classifiers Learned	FFRR	FR	RF	Weighted Average
Random Forest	0.689	0.93	0.722	0.754
SVM - PolyKernel	0.691	0.917	0.74	0.756
Simple Logistic	**0.741**	**0.941**	**0.774**	**0.796**

7. Count vs. Score representations with AthaMap matrix-based motifs (family level):

   Tables 5.8 and 5.9 report performance of classifiers trained on 21 AthaMap matrix-based motifs at family level, pertaining to count and score representations of motifs, respectively. A closer look at values in these tables identify count representation to be superior.

Table 5.8: *Cross-validation results with AthaMap matrix-based motifs (family level) using count representation*

Classifiers Learned	FFRR	FR	RF	Weighted Average
Random Forest	0.726	0.933	0.758	0.783
SVM - PolyKernel	0.708	0.916	0.786	0.776
Simple Logistic	**0.736**	**0.935**	**0.791**	**0.796**

Thus, our results show that, irrespective of how we deal with motifs found in the region of interest, whether we learn from motifs at factor level or from motifs at family level, counting occurrences (count representation) is a better way of training classifiers as compared to averaging over occurrence scores (score representation).

We also experimented with a *binary representation*, where attributes in the feature vector are marked as 0 or 1 indicating the absence or presence of the corresponding motif, respectively. This representation is even worse. In decreasing order of perfor-

Table 5.9: *Cross-validation results with AthaMap matrix-based motifs (family level) score representation*

Classifiers Learned	FFRR	FR	RF	Weighted Average
Random Forest	0.699	0.932	0.731	0.762
SVM - PolyKernel	0.675	0.918	0.722	0.744
Simple Logistic	**0.735**	**0.938**	**0.766**	**0.791**

mance, count > score > binary representations.

8. Feature selection over AthaMap and PLACE motifs (factor level):

   As expected, when a very small number of features are available, the performance of the classifiers is not very good. However, with increase in the number of features, the AUC values increase. However, when too many features are added, the performance of the classifiers decreases or remains constant (Fig. 5.2).

   Table 5.10 shows the five most predictive motifs found in AthaMap and PLACE based on feature selection.

   Table 5.10: *Five most predictive motifs for both AthaMap and PLACE*

AthaMap Motifs	**PLACE Motifs**
CBF	GT1CONSENSUS
TBP	ARR1AT
GT-3B	POLLEN1LELAT52
NTERF2	DOFCOREZM
DOF2	GT1GMSCAM4

9. AthaMap family level motifs with GC content as an added feature:

   By comparing Tables 5.2 and 5.11, we can see that including the GC contents as an extra feature does not improve the performance of the classifier. Instead, it acts as noise, lowering the classifier performance.

10. AthaMap factor level motifs and "gene1-intergenic-gene2" lengths as extra features:

    By comparing Tables 5.1 and 5.12, we can see that motif features together with the extra features corresponding to the "gene1-intergenic-gene2" lengths result in an increase

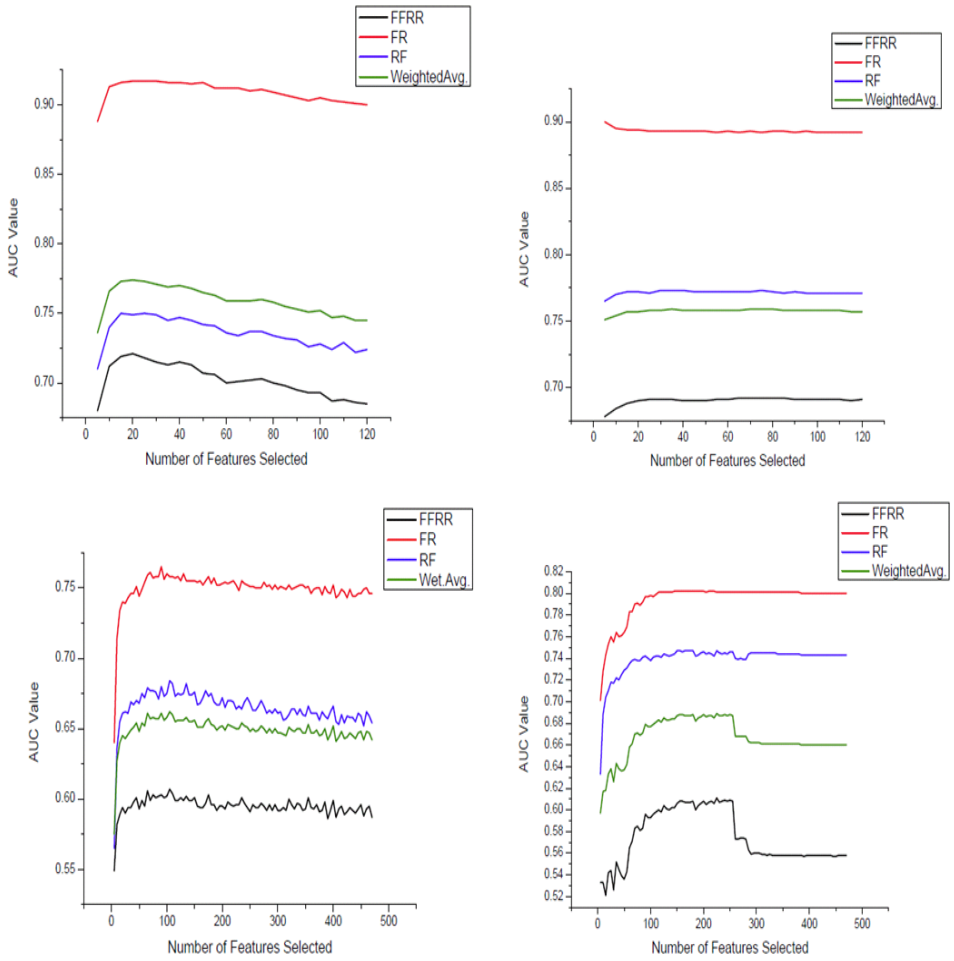

Figure 5.2: *The Area Under the ROC Curve as a function of the number of features selected for both Random Forests (left plots) and Support Vector Machines (right plots) using AthaMap (upper) and PLACE motifs (lower plots), respectively. Using a relatively small number of features (motifs), the classifiers achieve the best performance. As we add more and more features, the performance of the classifiers decreases or remains the same.*

Table 5.11: *Cross-validation results with AthaMap motifs (family level) and GC content as an extra feature*

Classifiers Learned	FFRR	FR	RF	Weighted Average
Random Forest	0.71	0.914	0.741	0.766
SVM - PolyKernel	0.685	0.894	0.775	0.756
Simple Logistic	**0.703**	**0.907**	**0.778**	**0.769**

in the classifier's performance.

Table 5.12: *Cross-validation results with AthaMap motifs (factor level) and gene1-intergenic-gene2 lengths as extra features*

Classifiers Learned	FFRR	FR	RF	Weighted Average
Random Forest	0.682	0.905	0.738	0.748
SVM - PolyKernel	0.696	0.895	0.804	0.769
Simple Logistic	**0.756**	**0.939**	**0.813**	**0.813**

11. AthaMap family level motifs and "gene1-intergenic-gene2" lengths as added features:

    As has been seen above, including an extra GC content feature to a set of abstract features (family level motifs) reduces the classifier's performance. The same is true when Tables 5.2 and 5.13 are compared.

Table 5.13: *Cross-validation results with AthaMap motifs (family level) and gene1-intergenic-gene2 lengths as added features*

Classifiers Learned	FFRR	FR	RF	Weighted Average
Random Forest	0.707	0.905	0.799	0.775
SVM - PolyKernel	0.729	0.939	0.773	0.789
Simple Logistic	**0.752**	**0.937**	**0.807**	**0.809**

12. Gene pair vs. region specific motifs (for AthaMap):

    Tables 5.14 and 5.15 present 10-folds cross-validation results with AthaMap factor and family level region specific motifs, respectively.

    Comparing Tables 5.1 & 5.14 (factor level), and 5.2 & 5.15 (family level), we found that at factor level, for both gene pair and region specific motifs, the performance of the Random Forest classifier was similar. But in the case of region specific motifs, the

SVM and Simple Logistic classifiers perform by 15% and 5% better than their gene pair counterparts, respectively. At family level, the SVM performance was similar for both types of features. However, Random Forest and Simple Logistic classifiers showed a 10% increase in AUC values with gene pair features.

Table 5.14: *Cross-validation results with AthaMap motifs (factor level) pertaining to different regions (gene1-intergenic-gene2)*

Classifiers Learned	FFRR	FR	RF	Weighted Average
Random Forest	0.682	0.905	0.738	0.748
SVM - PolyKernel	0.696	0.895	0.804	0.769
Simple Logistic	**0.756**	**0.939**	**0.813**	**0.813**

Table 5.15: *Cross-validation results with AthaMap motifs (family level) pertaining to different regions (gene1-intergenic-gene2)*

Classifiers Learned	FFRR	FR	RF	Weighted Average
Random Forest	0.729	0.936	0.773	0.789
SVM - PolyKernel	0.707	0.905	0.799	0.775
Simple Logistic	**0.752**	**0.937**	**0.807**	**0.809**

5.2 The Unbiased Approach

In the previous set of experiments, we collect motifs from biological databases and learn classifiers to solve prediction problems. In the unbiased approach, we assume that no motifs are available, and we use a sliding window approach to enumerate all possible k-mers (or motifs). Given the large number of features generated (Table 4.2) and, consequently, the increased amount of training time needed for each classifier, we will only report experimental results for the Random Forest classifier.

1. Cross-validation results when learning from separate k-mers:

   Table 5.16 reports the performance of the classifier trained over different k-mers, with number of features in each row being equal to those mentioned in the Table 4.2. As can

be seen, even when learning from all possible different k-mers, the classifier's performance is poor (55-65%) as compared to 88% with features from biological databases. Moreover, the best performance is obtained when the classifier is trained with 3-mers.

Table 5.16: *Cross-validation results when learning from different k-mers as separate data sets. Results shown for the random forest classifier.*

k-mers	FFRR	FR	RF	Weighted Average
3-mer	**0.567**	**0.68**	**0.596**	**0.601**
4-mer	0.559	0.667	0.591	0.592
5-mer	0.556	0.659	0.575	0.585
6-mer	0.552	0.654	0.574	0.581
7-mer	0.543	0.636	0.565	0.57
8-mer	0.545	0.638	0.558	0.57

2. Cross-validation results when learning from top-ranked separate k-mers (feature selection):

   To improve the results from the previous experiment, we do feature selection on k-mers. Table 5.17 shows AUC values as reported by the classifier. We see that, after removing irrelevant motifs using feature selection, the improvement in performance is not significant. In various k-mers, the performance was either similar or marked decimal increase in the AUC value. Instead of treating each set of k-mers separately, we should try combining all k-mers into one set, as biological databases usually consist of motifs of variable length.

Table 5.17: *Cross-validation results when learning from top-ranked separate k-mers. Results shown for the random forest classifier.*

k-mers	FFRR	FR	RF	Weighted Average
3-mer	**0.567**	**0.68**	**0.596**	**0.601**
4-mer	0.559	0.667	0.591	0.592
5-mer	0.562	0.661	0.58	0.59
6-mer	0.553	0.653	0.568	0.58
7-mer	0.541	0.633	0.571	0.57
8-mer	0.528	0.615	0.56	0.556

3. Cross-validation results when learning from top-ranked grouped k-mers (feature selection):

   Table 5.18 reports the performance of classifiers trained over 1350 grouped k-mers (see Table 4.3). Comparing Tables 5.17 and 5.18, it is clear that Simple Logistic shows improved performance. Including all k-mers as features definitely provides more information to the classifiers for learning. Although, the improvement is not satisfyingly significant.

   On the basis of transcription family level results from biological databases, we try to capture generalization via feature abstraction over 1350 k-mers, for a similar experiment.

Table 5.18: *Cross-validation results with top-ranked separate k-mers.*

Classifiers Learned	FFRR	FR	RF	Weighted Average
Random Forest	**0.564**	0.665	0.587	0.593
Simple Logistic	0.548	**0.688**	**0.624**	**0.599**

4. Cross-validation results learning from "best cut" clusters of grouped k-mers (from feature abstraction over selected motifs):

   Fig. 5.3 shows a graph correlating "number of clusters" and "AUC value". We start at the root node with one cluster, with the increase in number of clusters, AUC value increases and the performance reaches maximum at a cut with 1200 clusters. Further increase in cluster count results in lower AUC value. Hence, instead of 1350 features, we trained the classifier better with 1200 clusters (Table 5.19) and less training time.

   There are significant differences in classifier performance when trained with AthaMap family level motifs vs. k-mers clustered together based on sequence similarity. We assumed k-mers are unknown motifs and a way to group them could be to check similarity between sequences. But, with lower AUC value as compared to known motifs, it is evident that sequence similarity fails to capture biological significance in concept clusters. On the other hand, AthaMap motifs were grouped depending on

transcription families, with motifs sharing similar biological properties in each family.

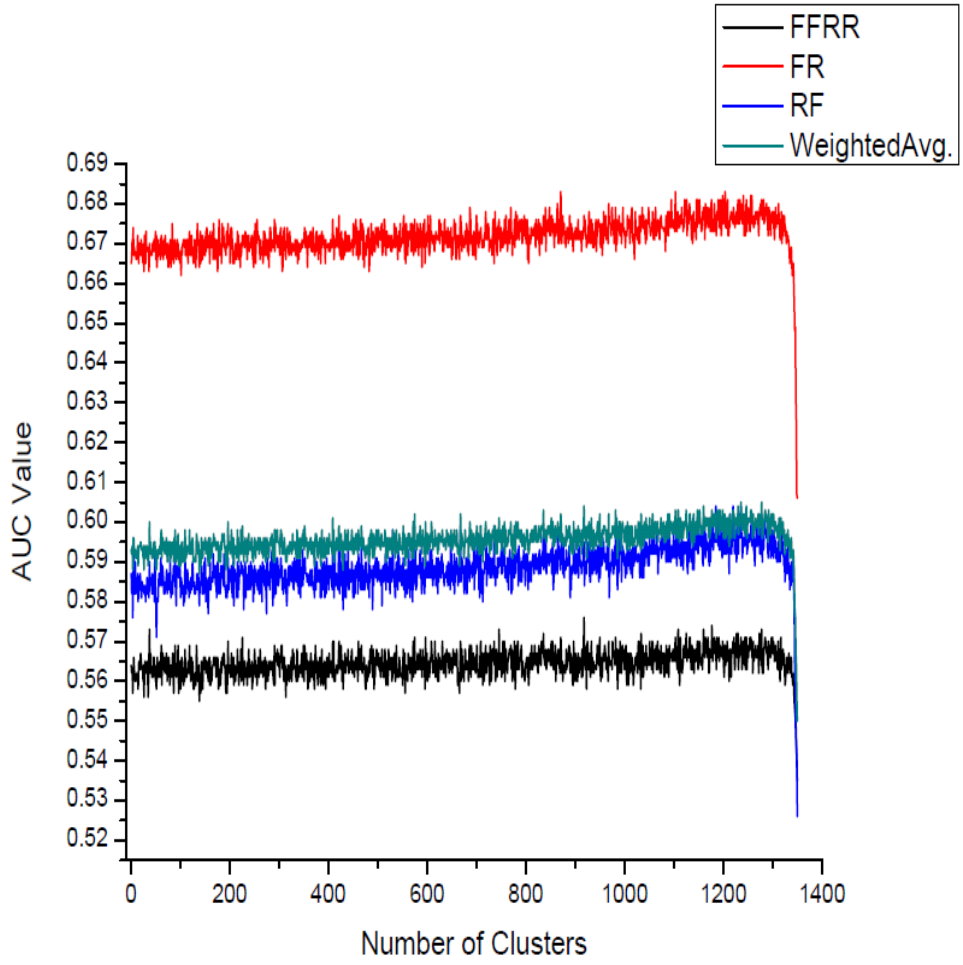

Figure 5.3: *Feature abstraction graph showing correlation between number of clusters (as features) and AUC value.*

5.3 Classifier Confidence Approach

Performance of classifiers in unbiased approach are worse than their respective counterparts in biological databases approach. We collected k-mers of variable lengths and these k-mers also include binding sites not yet reported in the databases. The results are poor because the variable length sites highly overlap. To resolve this issue, we train Random Forest classifier over top-ranked separate k-mers and build k-mer models for predicting class labels for test instances. 75% is training data and remaining 25% is test data. For each test instance,

Table 5.19: *Cross-validation results with best cut grouped k-mers clusters as features*

Classifiers Learned	FFRR	FR	RF	Weighted Average
Random Forest	0.571	0.683	0.597	0.603

k-mers models give class distribution for predicted class label of the given test instance. For each test instance, we select the class distribution that correctly predicts class label and has the highest classifier confidence for that label. Fig. 5.4 illustrates this approach. AUC values reported for this approach, for each class label are: **FFRR = 0.775**, **FR = 0.79** and **RF = 0.74**. So far, these are the highest performance measures with unbiased k-mers. We believe the results are better with this approach, as we use one classifier per length and predict the most "confident" class, in other words, only one length is used, the others are dropped. This partially takes care of the overlap between motifs (specifically, the case when one motifs is completely included in another). Thus, no classifier will have fully overlapping motifs.

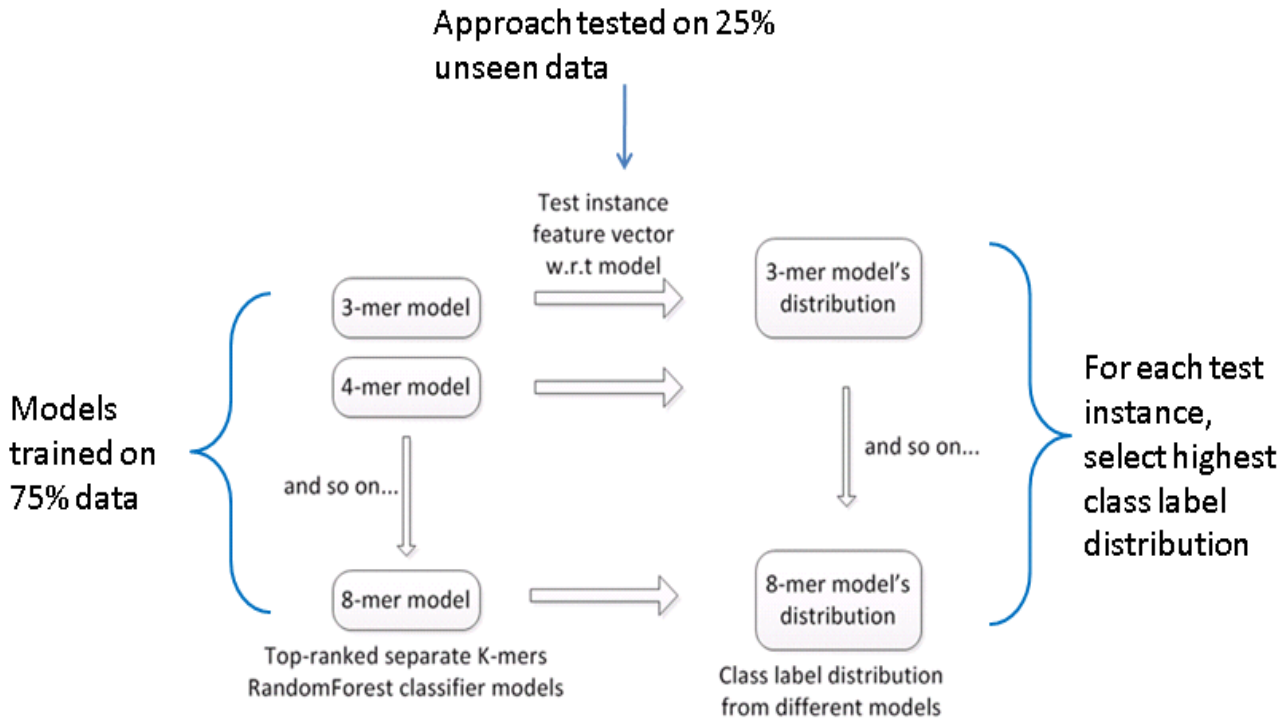

Figure 5.4: *Predicting the class label of test instances with the highest confidence k-mer model.*

Chapter 6

Related Work and Discussion

This chapter provides a review of the research related to the work presented in this thesis. We briefly discuss different motif finding algorithms other than Paster (used in AthaMap) and Signal Scan (used in PLACE). We also talk about research work where regulatory elements have played a key role and comment on the concept outlined in [Liu et al., 2008], closely related to our "classifier confidence approach".

Unraveling the mechanisms that regulate gene expression is a major challenge in biology. An important task in this challenge is to identify regulatory elements in DNA. Technical advances in genome sequence availability and high-throughput gene expression analysis have fostered the development of computational methods for motif finding [Das and Dai, 2007]. Current motif finding algorithms use phylogenetic footprinting or promoter sequences of coregulated genes or integrate both the approaches to find overrepresented binding sites. Algorithms discussed in this chapter correctly report motifs that were previously detected through lab experiments, and also find novel motifs. These algorithms work well with lower organisms DNA sequences (such as *Yeast* or *Arabidopsis*), but perform poorly in higher organisms [Das and Dai, 2007].

Helden et al. [1998] developed the motif finding algorithm *Oligo-Analysis* based on string-based exhaustive enumeration of motifs by counting and comparing oligonucleotide frequencies. String-based methods are well-suited for short motifs and generate huge collection of the same from DNA sequences. However, since motif positions are weakly con-

strained, some post-processing and clustering is needed to avoid problems arising from many spurious motifs. We follow a similar approach with k-mers and our results support the argument that grouping motifs into concept-clusters provides a better learning experience for classifiers. Another version of Oligo-Analysis algorithm (*Dyad-Analysis*) searches for spaced dyad motifs, where the length of spacer is varied between 0 and 16 and a motif is scored based on combined score of the two conserved parts.

Liu et al. [2004] developed the algorithm *FMGA* based on the concept of genetic algorithms. In contrast to string-based methods, they use probabilistic sequence model where motifs are represented by position weight matrix or PSSM - an approach similar to reporting matrix-based motifs in AthaMap. Probabilistic methods require few search parameters but rely on probabilistic models of the regulatory regions [Das and Dai, 2007]. These algorithms are biased to finding longer motifs.

AlignACE [Roth et al., 1998], ANN-Spec [Workman and Stormo, 2000], GLAM, Improbizer, Gibbs sampling, MEME [Bailey and Elkan, 1995], MotifSampler [Thijs et al., 2002], Bioprospector, MDScan, and QuickScore are other popular motif finding algorithms used in the research community. Some of these algorithms are efficient in identifying small motifs while others report longer motifs. Since little is known about transcription factors and their binding sites, these tools are designed for the discovery of novel regulatory elements, without assuming any prior information. Statistically overrepresented motifs in the "region of interest" is the desired output. However, performance of a single tool depends upon the type of data set used for evaluation, number of runs and various other tool-specific configurations. Relying on a single tool for our prediction problems is rather naive.

Ensemble algorithm by Hu et al. [2006] is a clustering-based algorithm that combines multiple predictions from multiple runs of five base component algorithms mentioned above. It takes advantage of promising predictions of component algorithms. As a result the overall performance of nucleotide level prediction is 22.4% more than stand-alone component

algorithms.

Holloway et al. [2008] target their research in the identification of transcription factors and the genes they regulate. They use features such as DNA patterns and gene expression experiments to identify true and false targets of transcription factors. Post-processing of Support Vector Machines based classifiers results and better feature ranking strategies have increased the overall performance to 86%. Their positive data set consists of known transcription factor binding sites published in the literature and negative data set consists of genes not bound by transcription factors, results from ChIP-chip experiments. "Using several genomic datasets a classifier is constructed for each transcription factor on a chosen set of features and then evaluated using a leave-one-out cross validation approach" [Holloway et al., 2008]. For each transcription factor, 100 classifiers are constructed, each using a random sub-sample of the negative set. A classifier built on the training set is evaluated using leave-one-out cross validation. In each cross-validation split, top 1500 features are selected, classifier is trained and then tested on the test set. The procedure is repeated 100 times and the net performance is the average of 100 cross-validation accuracies. For classifying a new target of a transcription factor, 100 classifiers are applied to the target gene's feature vector, and if the average probability of the gene being a target gene is greater than 0.5 then its a positive classification, negative otherwise.

Liu et al. [2008] proposed a time efficient feature extraction method to provide better and faster online search capabilities especially with image data. To identify correct image type, a classifier need not extract all features related to an image. For simple images, less features are adequate for high confidence prediction; however, for complex images more features are required. As a result, there is a tradeoff between classifier performance and feature extraction time cost, such that for each image instance, the overall test time cost is reduced in maximum subject to a condition that the classification performance is still acceptable. According to the authors, this approach is an upgrade to traditional feature selection that selects same subset of features for all instances. The confidence of Support Vector Machines

on a test instance is measured by the distance between the concerned instance and the decision boundary. The larger the distance, the more confident the classification [Liu et al., 2008]. For a new test instance, the first feature is extracted and the first classifier is fired. If the distance is larger than some threshold, the current classification is regarded as the result; otherwise next feature is extracted and the next classifier is invoked. We use a similar approach in Section 5.3, the difference being, each test instance is classified using different classifier models (learned from different k-mers) and the final class label is the one pointed out by the highest class distribution value of the actual class.

Chapter 7

Conclusion

In the recent years there has been enhancements in genome sequencing and high-throughput gene expression analysis technologies that has led to the availability of tremendous amount of DNA data. Also, there has been improvements in machine learning algorithms that can help researchers to quickly analyze nucleotide sequences and extract relevant biological information. In addition to studying motifs in labs, nowadays computational approaches are implemented to find solutions to biological problems from different perspectives. One such problem that has kept researchers occupied is understanding the mechanisms of gene regulatory networks. Modeling approaches followed by researchers are wide and disparate. Some gene regulatory networks are modeled entirely using non-parametric approaches such as Bayesian or neural networks, while some others represent genes in parametric differential equation formats [Das et al., 2009]. Encouraged by the fact that transcription factors and their binding sites play significant roles in identifying functions of regulatory networks, in this thesis we presented a motif-based machine learning approach for the same. A spin-off from focusing on constructing regulatory network from *Arabidopsis thaliana* genes, we were interested to find motifs that are predictive of transcription patterns of consecutive genes across the genome. Understanding latter would help better understand former, as suggested by our collaborator Dr. Volker Brendel from Iowa State University.

Motifs collected from biological databases and k-mers have different binding sequences, they belong to different transcription families. Some are well studied in labs, while other

are yet unknown. Some are relevant for transcription prediction problems while others act as noise, as seen from feature selection and abstraction experiments. A careful examination of putative motifs can offer new insights into genomic research. We collected motifs from AthaMap, PLACE and unknown k-mers, analyzed them to find out:

- Count representation is more suitable for gene pair transcription pattern prediction problems.
- Motifs from "AthaMap and PLACE", generate better feature vectors as compared to motifs from "AthaMap" or "PLACE".
- Classifiers learned from AthaMap data, perform better than classifiers learned from PLACE data. Former being a more comprehensive database pertaining to regulatory elements found in the region of interest, classifiers performance are approximately 88%.
- Classifiers learn better when provided with family level as compared to factor level features. Similarly, grouping k-mers into concept-clusters, improves performance. But, we have to be careful while capturing biological significance of motifs grouped into clusters. "Sequence similarity" is not an efficient parameter for the same.
- Techniques such as feature selection and abstraction, help discard irrelevant attributes in the feature vectors, and group features into concept-clusters. In turn, providing better performing classifiers.

Chapter 8

Future Work

This chapter showcases several related problems that we would like to address in future work. They are briefly described in what follows:

- Biological significance of most predictive motifs:

  Table 5.10 enumerates most predictive motifs collected from AthaMap and PLACE. These motifs were filtered from feature selection and abstraction experiments. To verify our approach and identify biological importance of these motifs, as a future work, we plan to search published research papers that talk about these motifs with respect to their presence in gene regulatory networks. Not all motifs are studied in genetic labs but even if we are able to find some published proofs of important motifs, then that would suffice. This task may also point towards genes that might be excluded from regulatory network but in reality they have a role to play. Using gene referrals from published papers we can also construct a sub-regulatory network that can show gene-gene regulation links, next task highlights the same.

- Using Textpresso to construct partial gene regulatory networks:

  Textpresso is an open-source search tool, that indexes full published papers for multiple organisms, including *Arabidopsis*. Given an initial set of genes, using Textpresso, we will identify a bigger set of genes that have regulator-target relationship among themselves. Using another tool called Cytoscape, we will view the network in form

of graph where, nodes of the graph are genes from *Arabidopsis*. Since researchers have not yet studied all possible gene-gene regulatory links, the information collected from published papers might be incomplete. As a result, the networks that will be constructed will be incomplete. They will help us better understand our current knowledge in gene regulatory networks. We may even combine this approach with motif-based machine learning approach in order to combine incomplete networks.

- Collecting motifs from multiple resources:

  As per Chapter 5, no database is complete enough to train classifiers such that their prediction performance is 100% and the approach of combining motifs from multiple databases is promising. Tompa et al. [2005] and Hu et al. [2005] suggest to use multiple motif extraction resources (biological databases and motif finding tools) rather than relying on a single resource. Chapter 6 talks about Ensemble algorithm by Hu et al. [2006], that clusters motifs collected from different motif reporting tools, the algorithm could be a decent choice to work with. We plan to combine motifs from multiple sources, perform feature selection and cluster them in such a way that biological significance is well captured. Motifs set hence generated, would alleviate effects of missing information at its best.

- Using advanced clustering mechanisms:

  In unbiased approach, we noticed two facts. First, agglomerative clustering based on sequence similarity fails to captures biological relation between motifs, similar to one captured by transcription hierarchy while we grouped motifs at family level. We need to use more advanced clustering mechanisms and distance parameters to solve the issue. Second, classifier confidence approach has generated best predictive results so far in case of k-mers. Perhaps, a combination of both these mechanisms will open doors for yet untouched research prospects.

Bibliography

T. L. Bailey and C. Elkan. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning*, 21(1-2):51–80, 1995.

P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002. URL http://citeseer.nj.nec.com/berkhin02survey.html.

L. Blow, S. Engelmann, M. Schindler, and R. Hehl. Athamap, integrating transcriptional and post-transcriptional data. *Nucleic Acids Research*, 37(Database-Issue):983–986, 2009.

M. K. Das and H. K. Dai. A survey of dna motif finding algorithms. *BMC Bioinformatics*, 8(S-7), 2007. URL http://dblp.uni-trier.de/db/journals/bmcbi/bmcbi8S.html#DasD07.

S. Das, D. Caragea, W. H. Hsu, and S. H. Welch. *Computational Methodologies in Gene Regulatory Networks*. IGI Global, 2009.

E. Davidson and M. Levine. Gene regulatory networks. 102:4935–4935, 2005.

T. Fawcett. An introduction to roc analysis. In *Pattern Recognition Letters*, 2005.

C. Galuschka, M. Schindler, B. Blow, and R. Hehl. Athamap web tools for the analysis and identification of co-regulated genes. *Nucleic Acids Research*, 35(Database-Issue):857–862, 2007.

R. W. HAMMING. Error detecting and error correcting codes. *BELL SYSTEM TECHNICAL JOURNAL*, 29(2):147–160, 1950.

J. V. Helden, B. Andr, and J. Collado-Vides. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *Journal*

of Molecular Biology, 281(5):827 – 842, 1998. URL http://www.sciencedirect.com/science/article/B6WK7-45S4971-5X/2/7b3676682c25e1346cf36909fe2d7c48.

K. Higo, Y. Ugawa, M. Iwamoto, and H. Higo. Place: a database of plant cis-acting regulatory dna elements. *Nucleic Acids Research*, 26(1):358–359, 1998. URL http://dblp.uni-trier.de/db/journals/nar/nar26.html#HigoUIH98.

K. Higo, Y. Ugawa, M. Iwamoto, and T. Korenaga. Plant cis-acting regulatory dna elements (place) database: 1999. *Nucleic Acids Research*, 27(1):297–300, 1999. URL http://dblp.uni-trier.de/db/journals/nar/nar27.html#HigoUIK99.

D. T. Holloway, M. Kon, and C. DeLisi. Building transcription factor classifiers and discovering relevant biological features. 2008.

J. Hu, B. Li, and D. Kihara. Limitations and potentials of current motif discovery algorithms. *Nucl. Acids Res.*, 33(15):4899–4913, September 2005. ISSN 0305-1048. URL http://dx.doi.org/10.1093/nar/gki791.

J. Hu, Y. D. Yang, and D. Kihara. Emd: an ensemble algorithm for discovering regulatory motifs in dna sequences. *BMC Bioinformatics*, 7:342, 2006. URL http://dblp.uni-trier.de/db/journals/bmcbi/bmcbi7.html#HuYK06.

A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999. URL citeseer.ist.psu.edu/jain99data.html.

G. Karypis. CLUTO - a clustering toolkit. Technical Report #02-017, November 2003.

M. Lee and R. I. Mahato. Gene regulation for effective gene therapy. *Advanced Drug Delivery Reviews*, 61(7–8):487–488, 2009. Gene Regulation for Effective Gene Therapy.

F. M. Liu, J. P. J. Tsai, R. M. Chen, S. N. Chen, and S. H. Shih. Fmga: Finding motifs by genetic algorithm. In *BIBE*, pages 459–466. IEEE Computer Society, 2004. ISBN 0-7695-2173-8. URL http://dblp.uni-trier.de/db/conf/bibe/bibe2004.html#LiuTCCS04.

L. P. Liu, Y. Yu, Y. Jiang, and Z. H. Zhou. Tefe: A time-efficient approach to feature extraction. In *ICDM*, pages 423–432. IEEE Computer Society, 2008. URL http://dblp.uni-trier.de/db/conf/icdm/icdm2008.html#LiuYJZ08.

C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to information retrieval*. Cambridge University Press, first edition, 2008.

A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *Proceedings of the 15th National Conference on Artificial Intelligence*, 1998. URL http://explorer.csse.uwa.edu.au/reference/browse_paper.php?pid=233281312.

T. M. Mitchell. *Machine learning*. McGraw-Hill Companies Inc., 1997.

C. Needham, I. Manfield, A. Bulpitt, P. Gilmartin, and D. Westhead. From gene expression to gene regulatory networks in arabidopsis thaliana. *BMC Systems Biology*, 3:85, 2009. URL http://www.biomedcentral.com/1752-0509/3/85.

NLM. Handbook - help me understand genetics. *U.S. National Library of Medicine*. URL http://ghr.nlm.nih.gov/handbook/illustrations/dnastructure.

D. S. Prestridge. Signal scan: a computer program that scans dna sequences for eukaryotic transcriptional elements. *Computer Applications in the Biosciences*, 7(2):203–206, 1991.

F. P. Roth, J. D. Hughes, P. W. Estep, and G. M. Church. Finding dna regulatory motifs within unaligned noncoding sequences clustered by whole-genome mrna quantitation. *Nature Biotechnology*, 16:939–945, 1998.

M. Sarachu and M. Colet. wemboss: a web interface for emboss. *Bioinformatics*, 21 (4):540–541, 2005. URL http://dblp.uni-trier.de/db/journals/bioinformatics/bioinformatics21.html#SarachuC05.

T. Schlitt and A. Brazma. Current approaches to gene regulatory network modelling. *BMC*

Bioinformatics, 8(S-6), 2007. URL http://www.biomedcentral.com/1471-2105/8/S6/S9.

N. O. Steffens, C. Galuschka, M. Schindler, L. Blow, and R. Hehl. Athamap: an online resource for in silico transcription factor binding sites in the arabidopsis thaliana genome. *Nucleic Acids Research*, 32(Database-Issue):368–372, 2004.

N. O. Steffens, C. Galuschka, M. Schindler, L. Blow, and R. Hehl. Athamap web tools for database-assisted identification of combinatorial cis-regulatory elements and the display of highly conserved transcription factor binding sites in arabidopsis thaliana. *Nucleic Acids Research*, 33(Web-Server-Issue):397–402, 2005.

G. Thijs, K. Marchal, M. Lescot, S. Rombauts, B. D. Moor, P. Rouz, and Y. Moreau. A gibbs sampling method to detect overrepresented motifs in the upstream regions of coexpressed genes. *Journal of Computational Biology*, 9(2):447–464, 2002. URL http://dblp.uni-trier.de/db/journals/jcb/jcb9.html#ThijsMLRMRM02.

M. Tompa, N. Li, T. L. Bailey, G. M. Church, B. Moor, E. Eskin, A. V. Favorov, M. C. Frith, Y. Fu, W. J. Kent, V. J. Makeev, A. A. Mironov, W. S. Noble, G. Pavesi, G. Pesole, M. Regnier, N. Simonis, S. Sinha, G. Thijs, J. Helden, M. Vandenbogaert, Z. Weng, C. Workman, C. Ye, and Z. Zhu. Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology*, 23(1):137–144, January 2005. URL http://dx.doi.org/10.1038/nbt1053.

N. D. Trinklein, S. F. Aldred, S. J. Hartman, D. I. Schroeder, R. P. Otillar, and R. M. Myers. An abundance of bidirectional promoters in the human genome. *Genome Research*, 14 (1):62–66, January 2004. URL http://dx.doi.org/10.1101/gr.1982804.

Q. Wang, L. Wan, D. Li, L. Zhu, M. Qian, and M. Deng. Searching for bidirectional promoters in arabidopsis thaliana. *BMC Bioinformatics*, 10(S-1), 2009. URL http://dblp.uni-trier.de/db/journals/bmcbi/bmcbi10S.html#WangWLZQD09.

I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations (The Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, 1st edition, 1999. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/1558605525.

I. H. Witten, E. Frank, L. Trigg, M. Hall, G. Holmes, and S. J. Cunningham. Weka: practical machine learning tools and techniques with java implementations. In *Proceedings of the ICONIP/ANZIIS/ANNES'99 Workshop on Emerging Knowledge Engineering and Connectionist-Based Information Systems*, pages 192–196, 1999.

C. Workman and G. Stormo. ANN-spec: a method for discovering transcription factor binding sites with improved specificity. *Pac. Symp.Biocomput.*, 5:464–475, 2000. URL citeseer.ist.psu.edu/workman00annspec.html.

W. Zhang, J. Ruan, T. H. D. Ho, Y. You, T. Yu, and R. S. Quatrano. Cis-regulatory element based targeted gene finding: genome-wide identification of abscisic acid- and abiotic stress-responsive genes in arabidopsis thaliana. *Bioinformatics*, 21(14):3074–3081, 2005. URL http://dblp.uni-trier.de/db/journals/bioinformatics/bioinformatics21.html#ZhangRHYYQ05.