Syracuse University

SURFACE

Dissertations - ALL

SURFACE

December 2017

Associative Pattern Recognition for Biological Regulation Data

Yiou Xiao
Syracuse University

Follow this and additional works at: https://surface.syr.edu/etd

Part of the Engineering Commons

Recommended Citation

Xiao, Yiou, "Associative Pattern Recognition for Biological Regulation Data" (2017). *Dissertations - ALL*. 821.
https://surface.syr.edu/etd/821

This Dissertation is brought to you for free and open access by the SURFACE at SURFACE. It has been accepted for inclusion in Dissertations - ALL by an authorized administrator of SURFACE. For more information, please contact surface@syr.edu.

ABSTRACT

In the last decade, bioinformatics data has been accumulated at an unprecedented rate, thanks to the advancement in sequencing technologies. Such rapid development poses both challenges and promising research topics. In this dissertation, we propose a series of associative pattern recognition algorithms in biological regulation studies. In particular, we emphasize efficiently recognizing associative patterns between genes, transcription factors, histone modifications and functional labels using heterogeneous data sources (numeric, sequences, time series data and textual labels).

In protein-DNA associative pattern recognition, we introduce an efficient algorithm for affinity test by searching for over-represented DNA sequences using a hash function and modulo addition calculation. This substantially improves the efficiency of *next generation sequencing* data analysis. In gene regulatory network inference, we propose a framework for refining weak networks based on transcription factor binding sites, thus improved the precision of predicted edges by up to 52%. In histone modification code analysis, we propose an approach to genome-wide combinatorial pattern recognition for "histone code to function" associative pattern recognition, and achieved improvement by up to 38.1%. We also propose a novel shape based modification pattern analysis approach, using this to successfully predict sub-classes of genes in flowering-time category. We also propose a "combination to combination" associative pattern recognition, and achieved better performance compared against multi-label classification and bidirectional associative memory methods. Our proposed approaches recognize associative patterns from different types of data efficiently, and provides a useful toolbox for biological regulation analysis. This dissertation presents a road-map to associative patterns recognition at genome wide level.

ASSOCIATIVE PATTERN RECOGNITION FOR BIOLOGICAL REGULATION DATA

By

Yiou Xiao
B.E. Nanjing University, 2009
M.S. Syracuse University, 2011

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Computer and Information Science and Engineering

Syracuse University
December 2017

Copyright © Yiou Xiao 2017

All rights reserved

ACKNOWLEDGMENTS

First and foremost, I am fortunate to be advised by three impeccable advisors: Prof. Kishan Mehrotra, Prof. Chilukuri Mohan and Prof. Ramesh Raina, who have given me tremendous help, encouragement and support. I am also grateful to my dissertation committee members: Prof. Jian Tang, Prof. Jae Oh and Prof. Shikha Nangia for their time and effort in providing me with invaluable feedbacks on the dissertation.

During my pursuit of research, Prof. Sucheta Soundarajan and Prof. Tomislav Bujanovic also provided invaluable advises for my thesis.

I am also grateful to my Fiancée, Jieqing Hu, for the encouragement and support during the past years.

Most of all, I am grateful to my parents for all the support and unconditional love. It is impossible to finish my PhD without them.

TABLE OF CONTENTS

Acknowledgments **iv**

List of Tables **ix**

List of Figures **xi**

1 Introduction **1**

1.1 Big data era of bioinformatics 2

1.2 Biological Regulation 3

1.3 Problem Formulation 6

1.4 Overall Objectives in Association Pattern Recognition 7

1.5 Dissertation Outline 8

2 DNA sequence affinity analysis **11**

2.1 Background of Aptamer Analysis 12

2.2 Problem Formulation 13

2.3 Related Methods in Aptamer Discovery 15

2.4 Modulo Addition Based Efficient Aptamer Search 16

2.5 Empirical Running Time Analysis 21

2.6 Concluding Remarks 23

3 Gene Regulatory Network **24**

3.1 Background of GRN Inference 25

3.2 Related Approaches for GRN inference 26
3.2.1 Approaches to undirected GRN inference 27
3.2.2 Approaches to directional GRN inference 31
3.3 Cis-elements and binding sites 33
3.4 Methods 35
3.5 Experiments and Results 38
3.5.1 AGRIS database 38
3.5.2 ARACNe v.s. ARACNe + TFTS 41
3.6 Concluding Remarks 42

4 Function Specific Histone Modification Pattern Recognition 43
4.1 Introduction 43
4.2 Methods 45
4.2.1 Related work 48
4.2.2 HiPSiS: Histone Profiling by Significance Score 51
4.2.3 Gene function prediction 52
4.3 Performance of simulated data 53
4.3.1 Pattern recognition performance 53
4.3.2 Real Datasets 55
4.3.3 Verification of Predicted Candidates for Specific Functions 56
4.3.4 Correlation between pattern scores and global ratios 57
4.3.5 Function-specific patterns predicted by HiPSiS 58
4.3.6 Evaluation of HiPSiS by gene function classification 59
4.3.7 Quality of Function Labels and Performance of HiPSiS 59
4.4 Concluding Remarks 60

5 Efficient combinatorial pattern recognition of series data 62
5.1 Introduction 63

5.2 Previous work 63
5.3 Pattern of Histone Curves 65
5.3.1 Similarity measures for time series 66
5.3.2 Algorithm for shape based pattern recognition 68
5.4 Comparison of Series Similarity Measures 73
5.5 Histone Curve Shapes 75
5.6 Case Study in Histone Modification Shape Patterns for flowering-time Genes 77
5.7 Concluding Remarks 78

6 Catworks: Associative pattern recognition between combinations of categorical data 80

6.1 Introduction 81
6.2 Problem Formulation 81
6.2.1 Categorical data 81
6.2.2 Data transformations 82
6.2.3 Objective 83
6.2.4 Related works 84
6.3 The Algorithm 86
6.4 Performance Evaluation 90
6.4.1 Synthetic Data Generating 91
6.4.2 Catworks performance analysis 91
6.4.3 Performance comparison 93
6.4.4 Evaluation on real dataset 96
6.5 Concluding Remarks 96

7 Conclusion 98

7.1 Summary 98
7.2 Future Research Problems 100

Appendices **103**

A **104**

A.1 Proposed Sub-Classes of Flowering Genes 104
A.2 Siginficant Combinations of Histone Modifications for Functions 111
A.3 Extended Study of HiPSiS with Other Labels 112
A.4 Details of Haar Transformation 113

References **115**

LIST OF TABLES

2.1 Categories of reads. 14

2.2 Numeric values assigned to nucleotides 18

2.3 The modulo-4 operation between nucleotides 19

2.4 Algorithm 1 running time (in seconds) w.r.t. the hash size and $|S|$ (2 million reads v.s. 8 million reads); t^H, t^E denote running time of Hamming distance and edit distance measures for quality control. 21

2.5 Time required by Algorithm 2 with respect to hash size with $|S| = 2M$. . . 22

3.1 t-test of Confirmed and unconfirmed edges.TFTS assigns significantly higher scores for confirmed edges compared with unconfirmed edges especially when sample size is big enough $(n_c + n_u > 30)$ 40

3.2 List of predicted new targets for gene AT1G24260 that agrees with AthaMap. # of matches refer to the number of matched fragments in the promoter region by AthaMap, where + represents the forward strand and - represents reverse strand. Average score is the confidence level of matching according to AthaMap. The details of scoring metrics in AthaMap can be found in [15] 40

4.1 An example of ChIP-seq data set in bed format. 45

4.2 The processed data with labels. Under Histone Modifications: 1 and 0 represent presence and absence of modification respectively; Under Function Labels: 1, 0 represent f and $\bar{f}$; * denote unknown functionalities 48

4.3 Notations and Description used in this chapter 49

4.4 Top 5 patterns recognized by HiPSiS: results are based on the average score of repeated experiments on 20 randomly simulated datasets with the same planted pattern [2,7,8] 55

4.5 Overview of TAIR gene GO annotation dataset 55

4.6 Verification of 15 genes with high scores $S_{\text{flowering-time}}$, predicted to have the functionality of "flowering-time". "ND function" means that the biological function of gene NF-YC1 related to flowering-time is not determined yet . . 56

4.7 The correlation of scores obtained using HiPSiS and simple basket analysis with background ratios. 58

5.1 Agreement ratio (defined in Equation 5.3) between proposed methods and DTW for simulated data. $|\Sigma|$ denotes the size of the alphabet used in the symbolic aggregation step; P denotes the length of series after approximation. These results are the mean values of 10 trials. 74

A.1 The detailed list of genes with combination of shapes "LLL". 107

A.2 The detailed list of genes with combination of shapes "VVV". 110

A.3 Function-specific combinatorial histone modification patterns. Functions labeling (marked with asterisk) are defined and obtained from TAIR gene ontology database [7]. Top patterns are desired modifications in the specific function, whereas bottom combinations are considered as inhibitors. 111

LIST OF FIGURES

1.1 Illustration of Gene Regulatory Network Modeling: According to the central dogma of biology, DNA is transcribed into RNA, which are further translated to proteins (represented by the dashed line between different layers). Genes (on the lowest level) are connected if their downstream products interact with each other. GRNs serve as a simplified model for the complex regulation system. 5

2.1 a. Acyclic aptamer identification process. The blue sequences represent the adapters for the aptamer sequences (red). [54], b. SELEX process of aptamer discovery[84] 13

2.2 Hash table with chaining. 18

3.1 Illustration of different scenarios of binding sites: (a) one single regulator might need multiple binding sites; (b) each transcription factor has its own binding site; (c) one single regulator can bind to any of the three binding sites. 34

3.2 Illustration of GRN inference using TFTS: (a) the input candidate network; (b) the database of cis-element occurrences; (c) gene-motif score; (d) refined GRN based on e^*_{ij}, where dashed edges are new proposed edges and the width of an arrow represents the confidence of the corresponding edge. . 36

3.3 The process of evaluating GRN inference accuracy between ARACNe and ARACNe+TFTS 42

4.1 Illustration of pre-processing of histone modification data. TSS is the starting point of transcription. h is the index of modification. 46

4.2 Pairwise correlation analysis of genes labeled with f_1 = flowering and f_2 = stress. 50

4.3 Emission probabilities in HMM model learned using ChromHMM with 5 hidden states. The $x-$axis shows elements of patterns (i.e., 0,1,2,3...9) and y-axis shows the indices of hidden states of the HMM. The intensity of each square (x, y) represents the level of confidence of including x in pattern y. (a) With $\alpha = 0.9$. States 2-4 clearly captured the planted pattern [2,7,8]. (b) When $\alpha = 0.5$. The planted pattern is not obvious anymore. . . 54

4.4 Classification Performance of HiPSiS and Simple Basket scoring for patterns 58

4.5 Classification Performance of ratios and simple basket transformed input data 59

5.1 Each plot shows the raw enrichment level data for each modification for two genes: (a) CO and (b) Actin. The x-axis is the position of modification relative to TSS of each gene. The red lines in each subplots shows the thresholds using p-value 10^{-8} 64

5.2 Illustration of shifted series data and original data. 67

5.3 Illustration of dynamic time warping method of time series comparison. The arrows represent the optimal alignment between sequences. 68

5.4 Wavelet decomposition step with 3 levels ($k = 3$) 69

5.5 Reconstruction with approximation using first 2 levels, the original series of length 8 is shortened to 2 (i.e., $k = 2$). 70

5.6 An illustration of the proposed approach with 5 series of length 10: (a) the original series; (b) approximated and normalized data; (c) discrete representations and number of occurrences; (d) the 2 centroids yielded by clustering. 72

5.7 Histone cluster centroids using approximated data ($P = 4$). Each curve represent a cluster centroid. X-axis shows the compressed indexes of original data (i.e., $j = 0, 1, 2, 3$.) 76

6.2 Performance when $N = 1$. 92

6.3 Performance when $N = 11$. 92

6.4 Perfect matching vs. Pattern ratio ($\beta = 0.15$) 93

6.5 Hamming loss v.s. Pattern ratio ($\beta = 0.15$) 94

6.6 Perfect matching vs. Pattern ratio ($\beta = 0.35$) 95

6.7 Hamming loss vs. Pattern ratio ($\beta = 0.35$) 95

A.1 ROC curves of HiPSiS with additional heat and salt stress. 112

CHAPTER 1

INTRODUCTION

The value of data analysis has become unprecedentedly recognized in the last decade. Nowadays, the great potential of data is appreciated by people from various backgrounds. From public relation experts to strategy makers; from big companies in silicon valley to scientists in fundamental sciences, people are devoting tremendous amount of attention, money and time to exploring its value. According to [21] the total size of bioinformatics databases has grown to 70 Petabytes in 2015 (10^9 MBytes); Twitter users generate 250 million posts every day [2].

The difficulty of storing, processing and analyzing big data has been recognized for a long time. However, the term "big data" gained great attention in recent years because of the big advancement in technologies [19, 47]. New infrastructures such as MapReduce, HDFS, Hadoop [82], NoSQL databases and GPU computation as well as deep neural networks algorithms such as LSTM [36] and CNN [53] rekindled the enthusiasm.

Associative patterns between sets of objects are of interest in many disciplines such as social networks, economics and biology. The goal is to discover the interactions or relations between sets of objects. Although many approaches have been proposed, most focus on interactions between single objects, considered using similar characteristics of objects. In this dissertation, we focus on associative patterns recognition in bioinformatics area.

Bioinformatics* is the ensemble of computational approaches to large-scale information analysis in biological data. It is now considered to be a self-contained branch of molecular biology, and helps researchers to better understand life systems; invent new diagnosis or treatment procedures; and design highly efficient medicines in target based therapies using data-centric techniques. Bioinformatics research accelerates the development of fundamental advances in biological hypothesis generation, data analysis and modeling, and provides tools for pharmaceutical, biomedical, chemical and even insurance companies. it encompasses a wide spectrum of topics that address questions about biological composition, structure, function and evolution of molecules, cells, tissues and organisms by computational methods that include mathematical modeling, machine learning and data mining. Biological regulation is defined as any process which modulates the frequency, rate or extent of biological processes, where computational approaches for recognizing interactions between objects (i.e., genes, RNAs, promoters, transcription factors and histone modifiers) [60] are crucially important in hypotheses generation and experiment design.

1.1 Big data era of bioinformatics

With the advent of highly efficient apparatus for sequencing, measuring and computing (Microarray [77], ChIP-seq [43] etc.), bioinformatics has entered the "Big Data Era" where large-scale and quantitative analyses of biological phenomena are made possible.

Researchers can quantify the dynamical phenotype changes and variations in biological systems with a fine-grained resolution via expression level profiling, systematically modeling the mechanisms of various types of regulation in terms of the relationships among different entities. It is important to understand the fundamental mechanism of biological

---

**(Molecular) bio-informatics: is conceptualizing biology in terms of molecules (in the sense of physical chemistry) and applying "informatics techniques" (derived from disciplines such as applied maths, computer science and statistics) to understand and organize the information associated with these molecules, on a large scale. In short, bioinformatics is a management information system for molecular biology and has many practical applications.* – Oxford English Dictionary regulation, to relate underlying causes with diseases and eventually treat them effectively [44] or culture new breeds of crops with immunity to diseases.

However, the tremendous growth in data size, dimensionality and variety pose new challenges in data usability. Various types of data (textual, time-series, sequence, categorical and numeric) are curated every year in large volumes of terabytes or even petabytes [†]at a continuously increasing rate. It is hence important and urgent to create new approaches to intelligently extract patterns and knowledge from large-scale heterogeneous data to accelerate hypotheses generation and experiment design in biological research.

1.2 Biological Regulation

Biological regulation is defined as any process which modulates the frequency, rate or extent of biological processes. Regulation mechanisms allow organisms to respond to various internal and external conditions/stimuli by maintaining the conditions under which constitutive processes remain viable [11]. Early studies of biological regulation can be traced back to the late 19th century when Bernard first proposed the idea of underlying mechanisms of regulation in living biological systems space [8]. A few decades later, with the development of molecular biology, biological regulation became the key to understanding the fundamental principles under cellular control. Multiple research works [39, 66, 40] proposed the idea of relationship networks and pathways between cells, which provided a foundation on which contemporary system biology was developed.

However, the terminology of "biological regulation" lacks a precise definition, and various different interactions between biological entities (cells, genes, RNA, proteins) are considered as parts of biological regulation processes; these are described below:

- **Cell Signaling:** This is the communication process that governs the basic activities of cells. It is critically important for a cell to function and respond appropriately to obtain

---

[†] According to statistics on EMBL-EBI, the total storage of bioinformatics data has reached 70+ petabytes, https://academic.oup.com/nar/article/44/D1/D20/2503123/The-European-Bioinformatics-Institute-in-2016-Data

information from the external environment, perceiving signals in the process of development, defense and tissue repair. Failures to respond correctly in human cell regulation lead to diseases such as cancer, autoimmunity or diabetes [89].

- **Protein-Protein Interaction (PPI):** This addresses the physical interactions between two or more protein molecules, required because proteins rarely act alone. The regulation of cells is carried out by molecular machines composed of many protein components. In the field of bioinformatics, protein-protein interactions are often modeled using a network (PPI network) whose nodes are proteins, and whose edges represent the various interactions between proteins [34]. Malfunctioning PPIs lead to diseases such as Alzheimer's disease and cancer.

- **Biological Pathways:** In this category, the actors of interactions are biological processes, functions or metabolism, instead of physical entities like DNA, RNA or proteins. Each process facilitates or inhibits others in the same pathway; [46] is a well known curated database of pathways in different organisms, assisting a systematical analysis of relationships between processes.

- **Gene Expression Regulation:** The coding regions in genes provide the blueprints for building RNA and eventually proteins in living organisms. However the genes don't express themselves equally. The RNA production rates are very different in different tissues or under diverse conditions, enabling cells to respond to internal and external conditions efficiently, although all cells are equipped with the same set of genes. However, our knowledge of how exactly gene expressions are regulated or controlled is very limited. Gene transcription regulation and histone modification are two very important mechanisms in gene expression regulation.

    1. In gene transcription regulation, one or more transcription factors which are products from certain regulatory genes bind to small fragments of sequences called Cis-regulatory elements (CREs) in the non-coding DNA and then activate or inhibit gene expression by increasing or decreasing the RNA production rate. The interactions

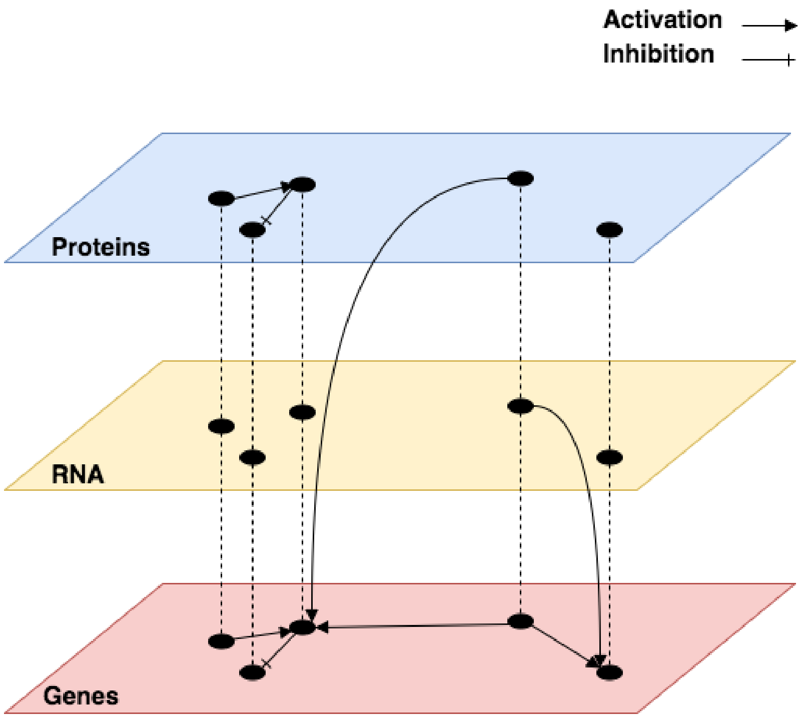


Fig. 1.1: Illustration of Gene Regulatory Network Modeling: According to the central dogma of biology, DNA is transcribed into RNA, which are further translated to proteins (represented by the dashed line between different layers). Genes (on the lowest level) are connected if their downstream products interact with each other. GRNs serve as a simplified model for the complex regulation system.

between genes are modeled as networks whose nodes represent genes, and whose edges represent either activation or repression (between proteins and genes layer in Figure 1.1). These interactions include a wide range of mechanisms that are used by cells to increase or decrease the production of specific gene products. In the last decade, many computational approaches were proposed to infer the links between genes. We propose an innovative approach, described in Chapter 3, which used CRE data to create the TF-CRE mapping score which is utilized as a measure to refine incomplete GRNs.

2. Another important type of gene expression regulation is histone modification. Histones are highly alkaline proteins found in eukaryotic cell nuclei that package and order the DNA into units called nucleosomes. Genes packaged in nucleosomes can

> only be expressed if they are available to transcription factors or RNA polymerase. Furthermore, histone code (combinatorial patterns of presence and absence of different histone modifications) is hypothesized to play a role in regulating gene expression in different biological processes.

In this work, we focus on computational approaches to address associative pattern recognition problems in the above two major regulation mechanisms in gene expression.

1.3 Problem Formulation

In this section, we formally define various associative pattern recognition problems at a high level. The concrete embodiment of each abstract concept is further described in chapters 2-6 where we elaborate each problem in detail. First, we introduce the following terminology:

An Object Universe is the collection of objects assumed to have the same type, such as the sets of all genes, textual labels, proteins or histone modifiers. Although our main focus lies in biological regulation, this concept can also be applied to other applications. For example, in social network analysis, users, hobbies or communities constitute different universes, respectively.

Let Ω_i denote the i^{th} object universe such that $\Omega_i \cap \Omega_j = \emptyset$ iff $i \neq j$. In general, there exist a large number of different universes in real-life applications. In this dissertation, we limit the number of universes to at most 2.

A Descriptor provides the mapping $\mathbf{D} : \Omega \rightarrow S$ where S is the feature space (of descriptor $\mathbf{D}$), such as time-series, geological locations, natural language, numeric values or any reasonable space. The concrete definitions of descriptor $\mathbf{D}$ and feature space S vary from application to application. For one object universe, there may exist more than one descriptors which describe different "facets" of the object. For example, for gene universe Ω_G, we have the following possible descriptors:

- Gene coding sequence: $\mathbf{D}_{CDS}(g) = x_1x_2\dots x_n$ where $x_i \in \{A, G, C, T\}$ defines the nucleotide sequence of gene g's coding DNA sequence;
- Gene expression level: $\mathbf{D}_{exp}(g) \in R^N$ defines the gene expression level (numeric values) at N time points.

The feature space S is defined as sequence data and numeric vectors, respectively, in the above examples. By allowing a flexible descriptor, we seek to manage heterogeneous data sources.

A Connector $\mathbf{C} \subseteq \Omega \times S$ is loosely defined as the relation between objects in one universe Ω and a feature space $S = \mathbf{D}(\Omega')$ where Ω and Ω' may be the same or different. A set describes the relation between objects and traits of other objects. For example: in binding sites analysis, we seek to learn the affinity of one object (genes, aptamers) to sequences which belong to other objects (genes); whereas in matchmaking algorithms, objects are users and $\mathbf{C}$ is then defined to be the users' preference of partners' traits (loyal, brave, handsome, etc.)

Baskets are transaction-like datasets of objects. $\mathbf{B}_i = \{X_i, Y_i, \dots\}$ is one entry of data where $X_i \subseteq \Omega_X$, $Y_i \subseteq \Omega_Y$ are considered to co-occur in basket data $\mathbf{B}_i$. For example, in market basket analysis, $B_i = \{\{\text{bread}, \text{apple}\}, \{\text{beer}, \text{wine}\}\}$ is a basket of objects from two universes: food and alcohol.

1.4 Overall Objectives in Association Pattern Recognition

Essentially, dynamic regulation is implemented by the interactions between different objects in organisms. The study of regulatory relations between genes, RNAs and proteins paves the way to understanding the fundamental clockwork in organisms. In our work, we propose efficient computational approaches for inferring associative patterns at different levels of regulation in biological systems, by data mining on multiple heterogeneous data sources. The objective is to accurately discover associative patterns between objects:

- Connector: To learn the connector $\mathbf{C}$ for a particular feature space S;
- $1:1$ patterns: To learn relations between objects from the universes: $A \subseteq \Omega \times \Omega'$;
- $n:1$ patterns: To learn relations between a combination of objects in one universe and a single object in another universe: $A' \subseteq \mathcal{P}(\Omega) \times \Omega'$, where $\mathcal{P}(\Omega)$ is the set of all combinations of universe Ω;
- $n:m$ patterns: To learn relations between combinations of objects from different universes. $A'' \subseteq \mathcal{P}(\Omega) \times \mathcal{P}(\Omega')$.

1.5 Dissertation Outline

This dissertation is organized as follows: In Chapter 1, we have discussed the recent advancement in bioinformatics which pose both challenges and opportunities. Then we introduce biological regulation and formal definitions in related problems described above.

In Chapter 2, we discuss an example of connector learning problem: DNA sequence affinity analysis. The objective is to find over-represented short sequences from a large NGS data set and the neighborhood of a sequence, within k mismatches. We introduce an efficient algorithm for protein-DNA affinity test by searching for over-represented DNA sequences using a hash function and modulo addition calculation.

Following this, in Chapter 3, we discuss an example of $1:1$ pattern recognition in gene regulatory network (GRN) link prediction where the nodes represent genes and edges represent regulation between genes. Most of the existing methods seek to find dependencies between two genes g_i, g_j by studying gene expression-level data as the only descriptor

($\mathbf{D}_{exp}$). As a result, most of the existing methods suffer from difficulties in causality learning, and is unable to prune indirect relations. Eliminating indirect regulations is critically important to gene regulation modeling because only an accurate GRN with direct regulation helps gain an understanding of the sophisticated and appropriate responses to external stimuli [62]. We propose a framework for refining weak networks based on transcription factor binding sites, thus improving the prediction performance.

In Chapter 4, we propose an algorithm for function specific histone combination pattern learning ($n : 1$ patterns). In histone modification analysis, it is assumed that different "histone codes" contribute to different biological process/functions in organisms. In our work, we first define a histone code to be subsets of all histone modifiers, and then study the basket data $\mathbf{B}_i = \{< H_i, f_i >\}$ where H_i is the set of modification present at record i and f_i is a binary label for a particular biological function. Then we associate each pattern with a score by comparing against its background frequency, and obtained the function specific patterns.

In Chapter 5, we show how Histone Profiling by Significance Score (HiPSiS) approach can be extended to more complicated patterns in which histone codes are defined as combinations of shapes of histone modifications (i.e., we treat each histone modification data as a time-series and study the locations as well as magnitudes of signal peaks). We introduce a procedure using series compression and symbolic aggregation methods for efficient clustering, and then we study the distributions of genes for different combinations.

In Chapter 6, we address the $n : m$ pattern recognition problems using an iterative algorithm. In our histone modifications analysis, we observed the existence of dependencies among "function labels": some genes are responsible for multiple roles in a biological system and some roles never co-exist on genes. We formulate the problem as a pattern association problem and propose an iterative algorithm to retrieve the hidden significant patterns between combinations of categorical values.

Finally, Chapter 7 provides the concluding remarks of this study and the future directions of research.

CHAPTER 2

DNA SEQUENCE AFFINITY ANALYSIS

In this chapter, we discuss one example of connector learning in DNA sequence affinity analysis. Given a pool of randomly generated short DNA sequences, a bead with a target protein is used to extract sequences from the pool. After extraction, the new sample is sequenced with high accuracy next generation sequencing (NGS) equipment. The objective is to efficiently and accurately find the "over-represented" sequence and its potential variations. In this project, we mainly focus on the application in aptamer affinity search: learning the connectors

$$\mathbf{C} : \Omega_{protein} \times \mathbf{D}_{sequence}.$$

We also used this approach in transcription factor binding site affinity learning, i.e.,

$$\mathbf{C}' = \Omega_{TF} \times \mathbf{D}_{promoter}.$$

In Section 2.4, we discuss an efficient sequence indexing and counting algorithm which is an example of creating connectors between different types of data formats. As a result, the connector quantifies the binding strength between proteins and target gene sequence. In Chapter 3, we use connectors to infer the relations between regulators and target genes by scanning target gene sequences for high-affinity sequences.

2.1 Background of Aptamer Analysis

Aptamers are oligonucleotides that bind to proteins, small organic molecules, or large molecules. Because of the high affinity and selectivity similar to antibodies, aptamers can be applied in fundamental research as capture agents, as well as in clinical applications such as cancer diagnosis and intervention [29, 31]. Additionally, aptamers are more robust because they can be preserved in a wider range of pH values and temperatures compared with antibodies. As a result, aptamer discovery has become an important topic in recent years.

Finding an efficient and reliable method of mapping nucleic acids with specific targets is one of the important challenges in biochemical engineering. SELEX [84] uses a cyclic, *in vitro* evolutionary method to find the desired nucleic acid aptamer. This method has been used as a standard approach in aptamer discovery for years. However, with the advent of high throughput deep sequencing technology [65], a new acyclic aptamer discovery method (illustrated in Figure 2.1.a) was proposed [54].

The objective of the novel aptamer discovery method is to efficiently search for the most over-represented (i.e., with the maximum number of copies) sequences after the acyclic enrichment process. Given the nature of this method, the output data contains millions or billions of reads. Consequently, an efficient approach is desired to quickly index and analyze the experimental datasets.

We propose an efficient algorithm* to process data from the acyclic aptamer discovery experiments to index and analyze the nucleic acid sequences in FASTQ format without compromising the quality of results. Compared with existing brute-force counting programs, our approach can reduce the running duration from hours to seconds thereby enabling researchers to quickly evaluate the quality of the experimental datasets as well as variants of the most highly represented aptamer using neighborhood search features.

---

*The source code of implementation of the algorithm is available at `http://sourceforge.net/projects/apthunter/files/`

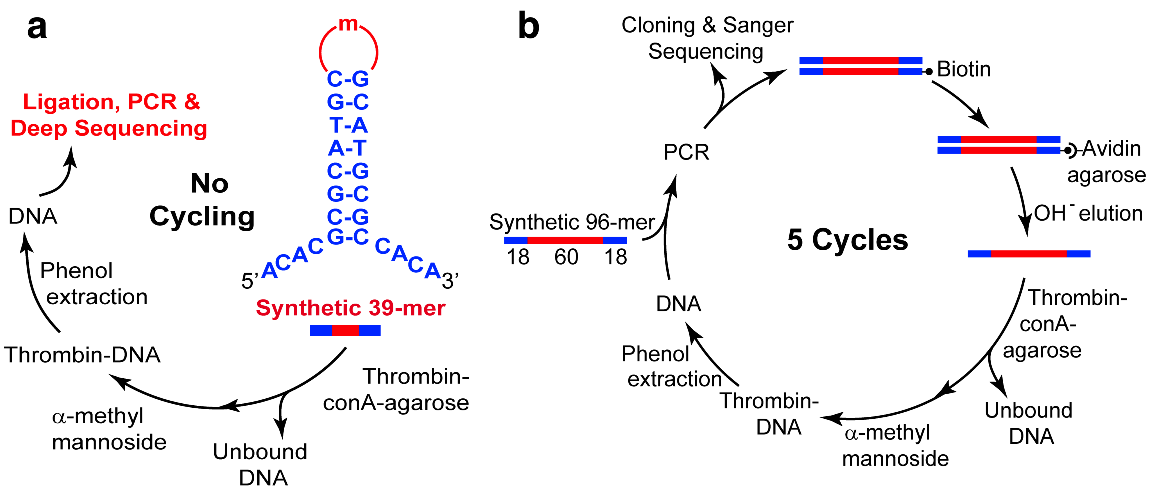


Fig. 2.1: a. Acyclic aptamer identification process. The blue sequences represent the adapters for the aptamer sequences (red). [54], b. SELEX process of aptamer discovery[84]

2.2 Problem Formulation

Let S be a set of N nucleic acid sequences of fixed length n on the alphabet $\Sigma = \{A, T, C, G\}$. Adapters are the two short nucleic acid sequences that are attached to the m-*mer*'s (aptamer) head and tail before sequencing (illustrated in Figure 2.1.a). They are referred to as *HEAD* and *TAIL*, of size h and t, respectively. These adapters are used to achieve quality control for a specific read (in the sense that reads without the desired head and tail can be viewed as noise sequences). Ideally, a "high-quality" read s^* in S has the following structure:

$$s^* = \textit{HEAD}\ s\ \textit{TAIL},$$

where s (the middle region) is a subsequence of size $m = n - h - t$. However, in practice, *HEAD* and *TAIL* may appear in s^* with some error; i.e., a typically observed sequence $s^* \in S$ is of the form

$$s^* = \widehat{\textit{HEAD}}\ s\ \widehat{\textit{TAIL}},$$

where sequences *HEAD*, and $\widehat{\textit{HEAD}}$ may differ from each other in one or more places; likewise for *TAIL* and $\widehat{\textit{TAIL}}$. Given a user defined tolerance threshold ϵ (> 0), if the distance

Categories	Description
Qualified Reads	$d(\mathit{HEAD}, \widehat{\mathit{HEAD}}) \leq \epsilon$ **and** $d(\mathit{TAIL}, \widehat{\mathit{TAIL}}) \leq \epsilon$
Candidate Read	$d(\mathit{HEAD}, \widehat{\mathit{HEAD}}) \leq \epsilon$ **or** $d(\mathit{TAIL}, \widehat{\mathit{TAIL}}) \leq \epsilon$
Uncertain Read	Otherwise

Table 2.1: Categories of reads.

between *HEAD* and $\widehat{\mathit{HEAD}}$ is less than ϵ, the approximation $\widehat{\mathit{HEAD}}$ is considered to be acceptable, otherwise not. A similar condition holds for *TAIL* also. Here the distance can be any reasonable distance, such as the Hamming-distance or the Edit-distance.

The sequences in S are categorized as qualified, candidate, and uncertain categories, as explained in Table 2.1. A read with both *HEAD* and *TAIL* within acceptable distances from the desired adapters is labeled as a qualified read; whereas if only one of them is within the acceptable distance from the desired adapter, the sequence is labeled as a candidate read; the rest are uncertain reads and are discarded.

Due to the nature of the experiments in aptamer search, we expect most of the s-strings in the middle region to be identical. Our goal is to find the common sequences and their number of occurrences. However, the frequencies of occurrence of some strings in S may be large due to the reason that these strings are made out of letters from Σ that appear in s^* randomly and by chance some of them may appear more often. But if a sequence occurs more often than expected by random chance, it is identified as "over-represented". Hence, the objective is to find all over-represented sequences and the associated numbers of occurrences. However, if this objective is not met, then we consider "highly-represented" sets under weaker conditions. Thus, for a given dataset S there are two possible objectives described below in descending order of preference, given the dataset S, *HEAD*, *TAIL*, and ϵ:

1. **Frequencies of the middle regions** – Find the frequencies associated with each distinct instance of middle region $s \in S$, separately for qualified and candidate sets.

The output consists of all sequences and associated counts in two hash tables: T_{Q} (Qualified) and T_{C} (Candidate).

If this objective is not met, i.e., if no significantly over-represented sequence is found in T_{Q} or T_{C}, then the following objective is addressed.

2. **Basic η-neighborhood** – In this case, to find a "highly-represented" sequence we consider the size of its neighborhood:

$$\mathcal{N}_\eta(s_i) = \{s_j |\ distance(s_i, s_j) \leq \eta\},$$

where η is a user specified parameter. If the size of the neighborhood set is large then the sequence is considered to be highly represented.[†]

Although the desired objectives, as described above, are different, the associated computational differences are negligible and essentially the same algorithm (with minor modifications) can be applied to address both objectives. These algorithms are described in the following section.

2.3 Related Methods in Aptamer Discovery

The current tool employs a simple exhaustive search and enumeration method [54]. It incrementally adds new sequences into an array. New entries are added if the sequence is not already present in the records and the count is incremented if the sequence is in the records.

A Perl script is used to identify sequence strings that closely match the 59- and 39-fixed regions flanking the degenerate bases (F5-m-F3). The match criterion (maximum allowed number of mismatches) is used to generate a file of *qualified reads* for sequences with the

---

[†] Sometimes neither of the above two objectives is met (i.e., neither we get an over-represented string nor an over-represented neighborhood) This is generally due to bad quality of data or the longer length of library region (e.g., for m40 library). In that case we conduct several experiments with identical setup to get several S-sets. In these S-sets we find the most common occurring sequence(s).

desired length of central m-bases. An m-*mer* count file is generated to obtain the number and rank for each unique sequence [54].

Because the nature of the experiments for aptamer search requires multiple experiments for different proteins under different conditions, the efficiency of sequence indexing and counting is crucial. However, the existing tool fails to be efficient due to the size of the dataset generated by next-generation parallel sequencing (NGS) technology [6], which tends to be very large.

2.4 Modulo Addition Based Efficient Aptamer Search

In this section, we discuss the details of our proposed approach. The current version of our program accepts two distance options: The Hamming distance between two sequences is easy to calculate but evaluation of the edit distance is more computationally expensive.

- **Hamming Distance function:** Given two sequences s and t, the Hamming distance [32] between them is defined as the

$$h(s,t) = \sum_{i=1}^{m} \mathbb{1}(s(i) \neq t(i)) \tag{2.1}$$

where $s(i), t(i)$ denote the i^{th} letters from s, t respectively; and $\mathbb{1}$ is an indicator function which returns 1 if its condition is satisfied. In essence, it calculates the number of mismatched letters between two sequences in a point to point comparison.

- **Edit-distance Evaluation:** Suppose we want to calculate the edit-distance, $d(s,t)$, between two strings s and t of lengths p and q respectively. According to Wagner-Fischer algorithm [91], the edit distance is defined as follows:

$$d'_{i0} = i \cdot w_{\text{del}} \qquad \text{for } 1 \leq i \leq p \tag{2.2}$$

$$d'_{0j} = j \cdot w_{\text{ins}} \qquad \text{for } 1 \leq j \leq q \tag{2.3}$$

$$d'_{ij} = \begin{cases} d'_{i-1,j-1} & \text{for } s_j = t_i \\ min \begin{cases} d'_{i-1,j} + w_{\text{del}} \\ d'_{i,j-1} + w_{\text{ins}} \\ d'_{i-1,j-1} + w_{\text{sub}} \end{cases} & \text{for } s_j \neq t_i \end{cases} \qquad \text{for } 1 \leq i \leq p, \text{for } 1 \leq j \leq q, \tag{2.4}$$

where $w_{\text{del}}, w_{\text{inst}}$ and w_{sub} are penalties for deletion, insertion and substitution, respectively. The edit distance between s, t is defined as $d(s,t) = d'_{pq}$.

The dynamic programing algorithm evaluates a table of size $O(pq)$. However, it is not necessary to evaluate this entire table for the following reason. Suppose that $d(k, \ell)$ denotes the edit distance between the first k characters of s and first ℓ characters of t. Then it is easy to verify that

$$d(k,\ell) \geq d(k',\ell'), \text{ for all } k' \in \text{Prefix}(k); \ell' \in \text{Prefix}(\ell).$$

Consequently, to improve the efficiency of the edit distance calculations, if $d(k',\ell') \geq \epsilon$, then there is no need to calculate $d(k,\ell)$ for longer sequences of k and ℓ.

- **Numeric Values and Hash Function**

To find the hash-value of a string s we use the following hash function[‡]:

$$H(s) = (\sum_{i=1}^{m} v(s_i) \times 4^j) \mod p, \tag{2.5}$$

[‡] In the case of DNA dataset we used 4 in the hash function. This integer is replaced by the size of the alphabet set, if it is different in other datasets.

where p is a prime number and represents the size of the hash table, and s_i is the i^{th} nucleotide of the read. The numeric values for $v(x)$ is defined in Table 2.2.

x	A	T	C	G
$v(x)$	0	1	2	3

Table 2.2: Numeric values assigned to nucleotides

- **Hash Table with Chaining**

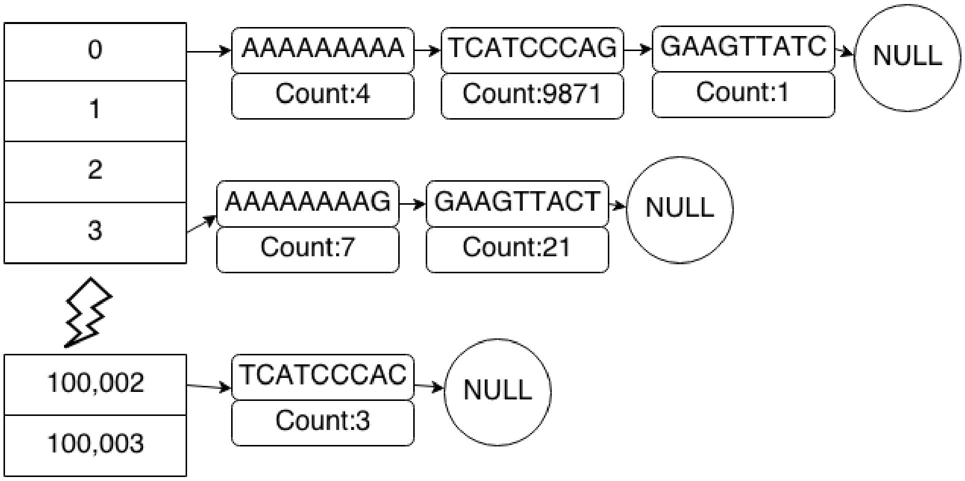


Fig. 2.2: Hash table with chaining.

The proposed algorithm makes use of a hash table, where conflicts are resolved by chaining, as depicted in Figure 2.2. The number of occurrences of a sequence is recorded in the linked list attached to the corresponding index. For ease of implementation, we first assign numerical values to nucleotides.

Often we allow minor perturbations in a s-sequence. Hence we need to know all neighbors of a given sequence that are within a neighborhood of (η) distance from it. Although such neighborhoods are required for all sequences that have high frequency in S, a procedure described below decreases computational effort significantly.

We use the procedure described in Algorithm 1 to store and index the number of occurrences of distinct middle regions. In the execution of this algorithm, two parallel processes

Algorithm 1 The Counting Algorithm

```
Require: $T$ is initialized to be a hash table of size $p$.

  function COUNT($S$, HEAD, TAIL, $\epsilon$)
      for all sequence $s^* \in S$ do
          $key \leftarrow H(s)$        ▷ $s$ is the middle region of $s^*$; $H(s)$ is defined in Equation 2.5
          if $d(\widehat{HEAD}, HEAD) \leq \epsilon \star d(\widehat{TAIL}, TAIL) \leq \epsilon$ then
              if $key \in T$ then
                  if $s \in T[key]$ then
                      $T[key][s] \leftarrow T[key][s] + 1$
                  else
                      $T[key][s] = 1$
              else
                  Create a linked list with one node $< s, 1 >$ on $T[key]$
      return $T$
```

are initialized with two hash tables T_Q, T_C representing qualified and candidate hash tables. Whenever a new sequence is encountered, the counting algorithm will check its quality by

$$d(\widehat{HEAD}, HEAD) \leq \epsilon \star d(\widehat{TAIL}, TAIL) \leq \epsilon,$$

where $\star$ is a logical operator. In case of T_Q we specify $\star \equiv \wedge$ which requires both ends of the sequence to be within at most ϵ distance from *HEAD, TAIL* respectively; whereas in case of T_C, we specify $\star \equiv \vee$, a weaker constraint.

- **Modulo-4 Addition Operation**

To reduce the computational complexity of η-neighborhood enumeration, we have adopted the modulo operation from [81]. This method is briefly described here. An addition operation is defined at nucleotide level as shown in Table 2.3.

$\oplus$	A	T	C	G
A	A	T	C	G
T	T	C	G	A
C	C	G	A	T
G	G	A	T	C

Table 2.3: The modulo-4 operation between nucleotides

The $\oplus$ operation can be easily extended for two different sequences u and v as well as when $\mathcal{S}$ is a set of sequences and v is a sequence, as shown:

$$u \oplus v = (u[1] \oplus v[1], u[2] \oplus v[2], \ldots),$$

and

$$\mathcal{S} \oplus v = \{u \oplus v | u \in \mathcal{S}\}.$$

Let o represent a string "AA. . . " of length m. The η-neighborhood of o is defined as

$$\mathcal{N}_\eta(o) = \{s' | \; d(s', o) \leq \eta\}$$

i.e., $\mathcal{N}_\eta(o)$ contains all possible strings s' such that the distance between s' and o is no larger than η. As a result, the η-neighborhood of a specific sequence s is easily obtained using $\mathcal{N}_\eta(s) = \mathcal{N}_\eta(o) \oplus s$. In other words, it is not necessary to evaluate the η-neighborhoods of frequent sequences, since these can be obtained by using the η-neighborhood of o.

Algorithm 2 The η-neighborhood Algorithm

```
Require: $T$ returned from Algorithm 1.
  function FINDN($S$, HEAD, TAIL, $\epsilon$, $\eta$, $\tau$)
      for all $s \in S$ such that $T[H(s)][s] \geq \tau$ do
          $\mathcal{N}_\eta(s) = \mathcal{N}_\eta(O) \oplus s$
          $W_s = T[H(s)][s]$
          for all $s' \in \mathcal{N}_\eta(s)$ do
              $W_s \leftarrow W_s + T[H(s)][s']$
      Sort $W_s$ in decreasing order of magnitude.
      return $W$
```

In Algorithm 2, the critical step is to find $\eta-$neighborhoods for all sequences, $\mathcal{N}_\eta(s_i)$. However, by using the modulo addition based method mentioned, we avoid the pairwise comparison of sequences. In our simulations, we need to initialize two processes for T_Q and T_C independently. The result of Algorithm 2 is a list of neighborhoods in descending

$\lvert S\rvert$ Hash size p	2×10^6		8×10^6	
	t^H	t^E	t^H	t^E
$10^5 (100,003)$	6	30	49	179
$10^6 (1,000,003)$	4	30	16	122
$10^7 (10,000,017)$	3	27	14	109

Table 2.4: Algorithm 1 running time (in seconds) w.r.t. the hash size and $|S|$ (2 million reads v.s. 8 million reads); t^H, t^E denote running time of Hamming distance and edit distance measures for quality control.

order of size. The top neighborhoods of the results are interpreted as the target aptamers (or their variants) with highest affinity to the protein on the extracting bead.

2.5 Empirical Running Time Analysis

The Human α-Thrombin dataset contains approximately 2.23 million sequences in FASTQ format, to which we applied our algorithm. (For a detailed description of the biochemical experiment settings, see [54].)

Using the Hamming distance function mode of our algorithm, we succeeded in discovering the leading target aptamer sequence as well as its η-neighborhood in under 6 seconds. However, if the Edit-distance function mode is used, the execution time was about 40 seconds, whereas the pairwise comparison based algorithm currently used takes 8280 seconds.

Both modes in our approach count the exact number of occurrences of each sequence; our approach has the same accuracy as the current software does.

Computational efficiency of Algorithm 1 is critical; it is the backbone of Algorithm 2. The performance of this algorithm is related to p, the size of the hash table. We experimented with several values of p and obtained the following results:

Table 2.4 and Table 2.5 show time required by Algorithms 1 and 2 respectively for different values of the size of the hash table and size of the data sets. All experiments were conducted using 3.06 GHz Intel i3 processors with 12GB, 1333MHz memory. The operating system was Mac OS X 10.7.4, and GCC compiler was used.

Size of the Hash-table	Time (Hamming distance)	Time (Edit distance)
10^5	7 seconds	39 seconds
10^6	5 seconds	32 seconds
10^7	4 seconds	30 seconds

Table 2.5: Time required by Algorithm 2 with respect to hash size with $|S| = 2M$.

From Table 2.4 and Table 2.5, it is clear that the proposed algorithms are sensitive to the size of the hash table. A hash table of too small size will suffer from a high collision ratio, which decreases the efficiency in counting and neighborhood discovery. In our experiments we have used hash-tables of sizes $10^5, 10^6$, and 10^7 and noted that running time for both counting and neighborhood searching decreases as the size of the hash table increases. However, the use of even larger hash tables is infeasible due to memory limitation, which could cause thrashing on a system with limited memory. Moreover, it will result in too many vacant slots in hash tables with empty records.

Distance measures are also important parameters. Obviously, the time complexity of edit distance is considerably higher than for the Euclidean distance. Is edit distance better than Hamming distance? The answer depends on the particular application. For some aptamer discovery applications, Hamming distance is a suitable choice due to the fact that insertions and deletions are rare. By contrast, in sequence alignment applications such as BLAST [3], edit distance measure is more preferable because the exact locations of fragments of interest are usually far from each other.

2.6 Concluding Remarks

This work addresses an approach to reduce the time complexity of sequence counting and the neighborhood discovery problem in aptamer searches. Our approach reduces the time by $1-3$ orders of magnitude and can handle large datasets, which is a significant improvement over the existing script-based technique. However, some additional modifications are likely to improve the performance. One in particular is to adopt a binary tree for each bin in the hash table. As a result, the initialization and look-up complexity can be reduced to $O(N \log M)$ and $O(\log M)$ respectively, where N is the number of sequences and M is maximum number of colliding sequences (i.e., with the same hash value $H(x)$).

In the worst case, if all sequences within the given dataset are distinct, then the space complexity will be high ($O(N)$). However, in the experiments we have performed so far, the sizes of most datasets can be reduced substantially since they contain many repeated sequences. Most of the experiments we have tested use less than 3 GB of memory. In future work, we can increase the algorithm's robustness in the worst case so that the tool's efficiency is not affected by the frequent page swapping of the operating system.

CHAPTER 3

GENE REGULATORY NETWORK

In this chapter, we discuss the "1:1" patterns learning in gene regulation. As discussed in Chapter 1, a gene regulatory network (GRN) describes interactions between genes, with directed edges representing the regulatory relationships (inhibition or activation) between genes. The absence of an edge between two genes implies that no direct relationship between them has been discovered. An important area of systems biology is the development of algorithms to infer the structure of GRNs, particularly with regards to the targets of transcription factors (TF). In real biological systems, the actual regulation occurs between a transcription factor (the protein product of a gene) and its target gene. However, a GRN is the simplified model whose nodes represent genes (both regulator and targets). As a result, the presence of an edge (g_i, g_j) means the transcription factor (TF) produced by gene g_i has a regulatory effect on gene g_j.

An important goal of gene regulatory network inference is to create a comprehensive map of interactions between TFs and genes. It is essentially a "1:1" pattern learning, where the universe of objects Ω is the collection of all genes in an organism. GRN inference seeks to find the association patterns $A \in \Omega \times \Omega$ by using one or more descriptors (i.e., promoter sequence $\mathbf{D}_{prom}$ or expression profile $\mathbf{D}_{exp}$).

In Section 3.1, we discuss the problem of GRN inference followed by a survey of existing approaches. In Section 3.4 we introduce a novel *Transcription Factor Target Scoring framework* (TFTS) and the motivation; then we evaluate the performance of TFTS as an add-on procedure to other GRN inference algorithms and study the improvement.

3.1 Background of GRN Inference

Gene regulation is one of the most important biological regulation mechanisms, which allows living organisms to adapt to their environment and maintain homeostasis. Genes are expressed and work in concert with each other to ensure the organism's fitness, survival and each cell's proper function. In order to maintain the appropriate functional outcome, each gene must be expressed at the proper time and in the right amount. The gene expression profiles for some genes are extremely similar in a given cell type [5], whereas the expression profiles of other genes vary considerably from cell to cell and from individual to individual, partly based on external cues and stresses. Genes do not work alone: every physiological phenomenon depends on the coordination between multiple genes, with the expression of some genes triggering or facilitating the expression of other genes relevant to the phenomenon of interest.

Experimental approaches, (such as ChIP-chip and ChIP-seq [43]) can be used to determine such relationships, and have achieved significant progress in identifying the target genes of a given gene. However, experimental work remains financially and technically difficult because thousands of TFs are involved in the process. Hence, computational approaches that analyze gene expression profiles have become useful in inferring regulatory relationship properties that help minimize the experimentation required. As high-throughput biological experiments have become prevalent in the recent decades, various computational approaches have been proposed to address the problem of GRN inference.

3.2 Related Approaches for GRN inference

Most of the existing computational approaches (for GRN inference) only use gene expression profiles data as the input. These approaches infer the relationships among genes by mining the interactions between genes in terms of gene expression levels. Based on the nature of the data, gene expression profile data can be roughly classified into two categories:

- **Condition-wise expression profile:** For each gene, the expression profile for a particular gene is represented by $X = \{X_1, X_2, \ldots, X_C\}$ where $1, 2, \ldots, C$ are indexes of conditions or even different experiments. In other words, the sequential order of observations is ignored and not used to infer relationships.
- **Temporal expression profile:** $X = \{X_1, X_2, \ldots, X_T\}$ denotes the expression levels at time points $1, 2, \ldots, T$. The order of observations is defined by the index of observation where X_i is observed earlier than X_j if $i < j$. Furthermore, given two time series (X, Y), X_i can only affect observation Y_j if $i < j$ [55]. The availability of data for this type of expression profile is very limited because of the difficulties in obtaining temporal expression data for higher eukaryotes [63].

In surveying related works, we find the following well known algorithms which focus on inferring gene-gene relations from single descriptor $\mathbf{D}_{exp}$ of each gene:

1. Algorithm for the Reconstruction of Accurate Cellular Networks (ARACNe) [63] focuses on the targets prediction using mutual information measures.
2. Ordinary Differential Equations (ODEs) and controlled perturbation are used in TSNI [25].
3. Bayesian network and Dynamic Bayesian network inference are employed in Banjo [97]
4. GeneNet [61] uses Graphical Gaussian Models to exclude indirect relationships.

Many other approaches along with the above methods can be roughly categorized into two classes based on their result:

- Undirected edges usually represent the similarity or symmetric relations between genes, where condition-wise expression profiles are used as descriptors of each object and these approaches quantify the relation strength between genes by measuring expression profile similarity. This approach is also called "co-expression" analysis.
- Directed edges are closer to real underlying regulation relations in biological systems. $A \rightarrow B$ represent the specific directional regulation of gene A on B. Temporal expression profiles are mostly used to infer such relations.

In the following sections, we will discuss the above two types of analysis in details.

3.2.1 Approaches to undirected GRN inference

Since co-expressed genes are likely to be functionally related [27], many methods were proposed to study co-expression relations between genes. Multiple similarity measures have been used to quantify proximity between two genes. Genes belonging to a cluster are considered to be functionally related to all the other genes in the same cluster. Such approaches can be applied to both aforementioned categories of expression data. Gene clustering via expression profile similarity is mostly applied to non-temporal expression data to discover non-directional relations, whereas temporal expression data provides more information about the directions of regulation. We will discuss two common types of similarity measures:

- **Correlation:** The most common measure is Pearson's correlation coefficient $r(X, Y) \in [-1, 1]$ defined as follows:

$$r(X, Y) = \frac{\sum_{k=1}^{C} X_k Y_k}{\sqrt{\sum_{k=1}^{C} X_k^2 \sum_{k=1}^{C} Y_k^2}} \tag{3.1}$$

$r(X, Y)$ quantifies the linear dependencies between genes expression levels, where a value of $+1$ represents total positive correlation which implies that gene X, Y may positively regulate (activating) each other, and -1 represents total negative correlation which indicates negative (inhibitory) regulation. This measure can be used with both condition-wise data and temporal data for co-expression analysis. Threshold τ is used to construct gene regulatory network by removing edges for which $r(X, Y) \leq \tau$, from a fully connected network $\Omega \times \Omega$. As a result, the remaining edges are relations with of a high confidence level.

- **Mutual Information:** Expression profiles X, Y can be considered as the observations for two random variables x and y. Mutual information for these random variables is denoted as $I(X, Y) = H(X) + H(Y) - H(X, Y)$ where $H(\cdot)$ denotes entropy. In the context of discrete values, we define

$$H(X) = -\sum_i p(X_i) \log[p(X_i)] \tag{3.2}$$

$$H(X, Y) = -\sum_i p(X_i, Y_i) \log[p(X_i, Y_i)]. \tag{3.3}$$

This measure is used in ARACNe which adopts the estimated joint distribution density function using a Gaussian kernel estimator [4] defined as:

$$p(x, y) = f(\vec{z}) = \frac{1}{nh} \sum_i^n K(\frac{\vec{z} - \vec{z_i}}{h})$$

where $\vec{z_i} = \{X_i, Y_i\}$ denotes the vector of two variables, h is the smoothing parameter and $K(\cdot)$ is a kernel function that satisfies $\sum_{\vec{z} \in R^2M} K(\vec{z}) = 1$. It is worthwhile to note that $p(x), p(y)$ are marginal distributions of the Gaussian kernel estimator $p(x, y)$. As in all similarity based edge inference algorithms, a threshold is used to determine whether an edge should exist between two genes, by checking the occurrence of $I(X, Y) \geq I_0$.

One concern with the similarity based approach is the false positive edges introduced by indirect regulation. For instance, consider three genes X, Y and Z, whose expression is governed by the regulation cascade:

$$X \to Y \to Z,$$

with no direct regulation from X to Z. But if we analyze the expression profiles of X and Z, it is possible to conclude that X regulates Z although there is no direct regulation. Another scenario is

$$X \to Y, X \to Z$$

where Y, Z are both target genes of regulator X. Then $sim(Y, Z)$ is high because both $sim(X, Y)$ and $sim(X, Z)$ are of high values and $sim(\cdot)$ is a symmetric measure of two variables.

ARACNe prunes indirect interactions by using the data processing inequality (DPI)[22] as the final step. For each triple of genes (X, Y, Z), ARACNe removes the edge e_{xy} if $I(X, Y) < min(I(X, Z), I(Y, Z))$.

GeneNet, a correlation based GRN inference algorithm, addresses this problem by using partial correlation, which measures the dependency between two variables in the presence of other variables:

1. In order to compute the sample partial correlations between two random variable samples X, Y, we need to solve the following two linear regression problems to get the residuals

$$W_X^* = \arg\min_W ||X - W * Z||^2 \tag{3.4}$$

$$W_Y^* = \arg\min_W ||Y - W * Z||^2 \tag{3.5}$$

   where Z is the matrix of samples from other variables (genes).

2. Calculate the residual of each variable:

$$r_X = X - W_X^* Z \tag{3.6}$$

$$r_Y = Y - W_Y^* Z \tag{3.7}$$

3. Finally, the measure of partial correlation (between X, Y with the presence of Z) $\hat{\rho}_{XY \cdot Z}$ between two samples is computed to be the correlation between r_X and r_Y.

- **Graphical Gaussian Model** is a simple method for inferring the network of linear dependencies among a set of variables. The objective is to correctly identify direct influences when a naive correlation approach is ineffective. The key idea is to use magnitudes of partial correlation as a measure of independence of any two genes, which is then used to distinguish indirect interactions. The gene expression levels are modeled using multivariate Gaussian distribution $D = [X, Y, Z \ldots] \sim \mathcal{N}_d(\xi, \Sigma)$, where ξ is the estimated mean (i.e., $\xi = [\bar{X}, \bar{Y}, \bar{Z} \ldots]$) and Σ is the estimated covariance matrix. If Σ is positive definite*, then $\Omega = \Sigma^{-1}$ is called the precision matrix. In the context of graph learning, the objective is to estimate the precision matrix $\tilde{\Omega} = \tilde{\Sigma}^{-1}$ using observed data X, where

$$\tilde{\Sigma} = \frac{1}{M-1} \sum_{i=1}^{M} (X_i - \xi)(X_i - \xi)^T \tag{3.8}$$

Because of the convenience of the precision matrix in representing partial correlations when D is a multivariate Gaussian distribution, GeneNet constructs edges using simple rule: $(a, b) \in E$ iff $\Omega_{a,b} \neq 0$. This type of network edge learning is based on the Gaussian Markov Random Field (GMRF) [49], which is applied to association learning [86] between variables distributed on a multivariate normal distribution. A MATLAB implementation of partial correlation based method is implemented [61] for genome-wide gene regulatory inference.

---

[*] An $n \times n$ complete matrix A is called positive definite if $z^T Az > 0$ for every non-zero column vector z of n real numbers.

3.2.2 Approaches to directional GRN inference

Directional regulatory relations between genes provide accurate models of how the underlying mechanism works. The regulators and targets are differentiated in this type of analysis so that the regulatory network is defined as a directed network instead of a symmetric network. Many different approaches were proposed in this field with various hypotheses of how exactly regulation affects the gene expression; two important related works are discussed below:

- **Dynamic Bayesian Network:** Bayesian network (BN) is a graphical model used to model the probabilistic relationships among a set of random variables $D = X_i$ where $i = 1...N$ are indexes of genes G. Formally, Bayesian networks consist of 3 components:

  - A set of random variables: In GRN inference, we model each gene's expression profile as one of the random variables.
  - The conditional dependencies between variables are represented by a directed acyclic graph $\pi(X) \subseteq G$, where if two variables X, Y satisfy $p(X|Y) = p(X)$, then the edge $Y \rightarrow X$ is not present in BN.
  - The conditional probability distribution for each variable $p(X|\pi(X))$ and joint probability of all genes are modeled by the equation:

$$P(X_1, X_2, ...X_N) = \prod_{i=1}^{N} P(X_i = x_i | X_j = x_j, X_{j+1} = x_{j+1} \ldots X_{j+l} = x_{j+l}) \quad (3.9)$$

  where the $l+1$ genes are the regulators (of gene i) whose expression values determine the value of gene i in the Bayesian network.

The learned structure of Bayesian network $\pi(X_i) \subseteq 1, 2, \ldots, N$ is considered as the GRN structure since $\pi(X_i)$ includes all the variables that affect X_i. $G =< V, E >$ where V is the set of all genes and directed edges E defines the statistical dependencies between variables.

The objective of *Banjo* [97] is to construct such a graph G so that the likelihood is maximized. In other words, it seeks to find the parameters of the Bayesian model that can fit best with the expression data D. In Banjo, they adopted Bayesian Information Criteria (BIC) as the fitness score

$$\text{BIC}(G||\theta, D) = \ln(M)k - 2\ln(\hat{L}),$$

where M denotes the length of observations; k is the number of free parameters for tunning; θ defines the conditional probability distributions; and G is the structure of the network.

$\hat{L} = p(D|\hat{\theta}, \hat{G})$ is the maximized value of likelihood function with model parameters $(\hat{\theta}, \hat{G})$, where $(\hat{\theta}, G) = \arg\max\limits_{\theta, G}(p(D|\theta, G)$.

Banjo initializes with a random network structure G, and iteratively refines the network G until no improvement can be achieved.

However, the BN model lacks self regulatory relations (i.e., the model is incapable of capturing the self-regulatory interactions of genes). Furthermore, BN cannot model delayed interactions. So Dynamic Bayesian Networks (DBN) [68] have been proposed to relate variables over adjacent time steps $t, t+1$:

$$\begin{aligned} &P(X_1(t+1), X_2(t+1), ...X_N(t+1)) \\ &= \prod_{i=1}^{N} P\left(X_i(t+1) = x_i | X_j(t) = x_j, X_{j+1}(t) = x_{j+1} \ldots X_{j+l}(t) = x_{j+l}\right) \end{aligned} \tag{3.10}$$

- **Ordinary Differential Equation:** This is another type of modeling technique where the change of a gene's expression level is determined by variations in other genes' expression levels.

$$\dot{X}_i(t+1) \triangleq [\frac{dx_i}{dt}]_{t+1} = f_i(x_1(t), x_2(t), \ldots, x_N(t); u, \theta) \tag{3.11}$$

In the case of sampled observations, the discrete model is defined as:

$$\Delta X_i(t+1) = f_i(x_1(t), x_2(t), \ldots, x_N(t); u, \theta) \tag{3.12}$$

where u is the external perturbation and θ is a set of parameters describing interactions among genes. If we constrain $f_i(\cdot)$ to be the linear function $f_i = WX(t)^T + u$, then $\theta = W$ is the coefficients weight matrix which indicates the inter-gene relations. In the corresponding GRN, there exists an edge $i \rightarrow j$ iff $|W_{ij}| > 0$. Furthermore, the sign of W_{ij} describes the type of regulation (activation or inhibition). TSNI [25] adopted this approach to infer the relevant transcription factors for TRP63 using temporal expression profiles.

The above computational approaches all suffer from inaccuracies because it is extremely difficult to differentiate indirect regulation and direct regulation based merely on expression profile. Hence we have proposed an innovative method TFTS [94] that takes preliminary networks inferred from expression profile dataset, and use transcription factor binding sites consensus to refine the network.

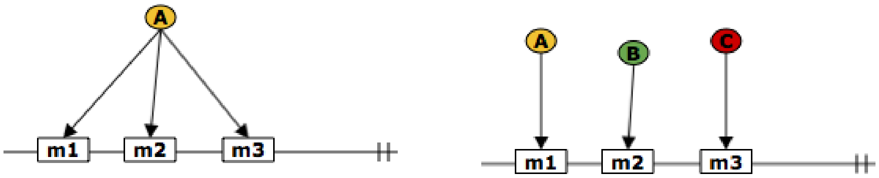
3.3 Cis-elements and binding sites

Cis-regulatory elements (CREs, cis-elements or motifs) are small fragments of sequences located on non-coding regions of genes. They are found in the vicinity of transcription starting site (TSS) of genes that they regulate. In the process of regulation, transcription factors have to physically bind to these small regions before regulation is initiated. We consider the presence and absence of cis-elements in the network refining algorithm, in order to overcome the difficulties in differentiating direct and indirect regulation.

We hypothesize that functional cis-elements occur more often in the target genes (of a given gene) than in all genes, and the increase in relative occurrence frequency is statistically significant. This hypothesis stems from the fact that TFs need to bind to cis-elements of the target gene to regulate its profile expression [76].

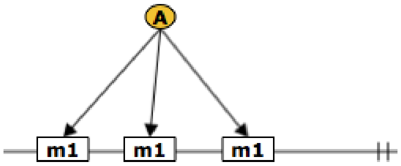
In sequence analysis of gene promoter regions, motifs can be classified into three main types.

1. Consecutive motif: no gaps are allowed in the subsequence.
2. Simple motif: no variable gaps are allowed in the motif. For example, CGS[11,11]SCG, where [11,11] means that there is a fixed "gap" of length 11 between the two consecutive motifs.
3. Structural motif: the gap length can vary in a range.



(a) Structural Motif

(b) Compound Factors in Regulation



(c) Redundant Motifs for the same TF

Fig. 3.1: Illustration of different scenarios of binding sites: (a) one single regulator might need multiple binding sites; (b) each transcription factor has its own binding site; (c) one single regulator can bind to any of the three binding sites.

- **Co-occurrences of cis-elements:** In real biological systems, target genes of a transcription factor usually have multiple co-occurrences of cis-elements in the promoter region.

This may be due to the fact that some transcription factors bind to a structured motif instead of a consecutive motif, as shown in (Figure 3.1-a). In other cases, the real binding site of the transcription factor is a complex motif with multiple gaps and may require multiple regulators to collaborate together, as illustrated in Figure 3.1- b. As a result the co-occurring motifs might be the real binding sites for multiple regulators [16].

- **Redundant occurrences of cis-elements:** One cis-element might appear multiple times in the promoter sequences, as shown in Figure 3.1 - c. This phenomenon is favored by evolutionary processes because it increases the robustness of the regulation system [71].

3.4 Methods

- **Problem Statement:** The input for TFTS consists of two parts:

1. A directed graph $G =< V, E >$ where V represents the collection of genes and E denotes the set of directed edges such that $e_{i,j} \in E$ if and only if gene j is a regulatory target of gene i. We use $N = |V|$ to denote the total number of genes.
2. A matrix $M = \{m_{ik}\}$ where m_{ik} denotes the number of occurrences of motif k in the promoter sequence of gene i.

G refers to the candidate network inferred using the algorithms mentioned in the previous section, and M is obtained either from a cis-elements database [24] or using binding sites discovery algorithms: HAMMER [80] or aptamer hunter [95]. The output of this process is a scoring function $f : G \times G \to R$ which quantifies the confidence of the predicted edge. As a result, we refine the input network by removing edges with low scores, and introducing new edges with high confidence (exceeding a threshold). The final result will be a refined network $G^* =< V, E^* >$.

- **Gene-motif scores:** The score of an edge is evaluated using the p-values associated with cis-elements in the target genes. The p-value is used in a standard statistical approach to calculate the probability of an event under the null hypothesis; a small p-value implies that the event is not governed by chance. We adopt the following notation:

  - $S_i = \{g_k | e_{i,k} \in E\} \subseteq V$ denotes the set of target genes of gene i. As shown in Figure 3.2(a), the target genes of gene A are C,D,E,F,G.

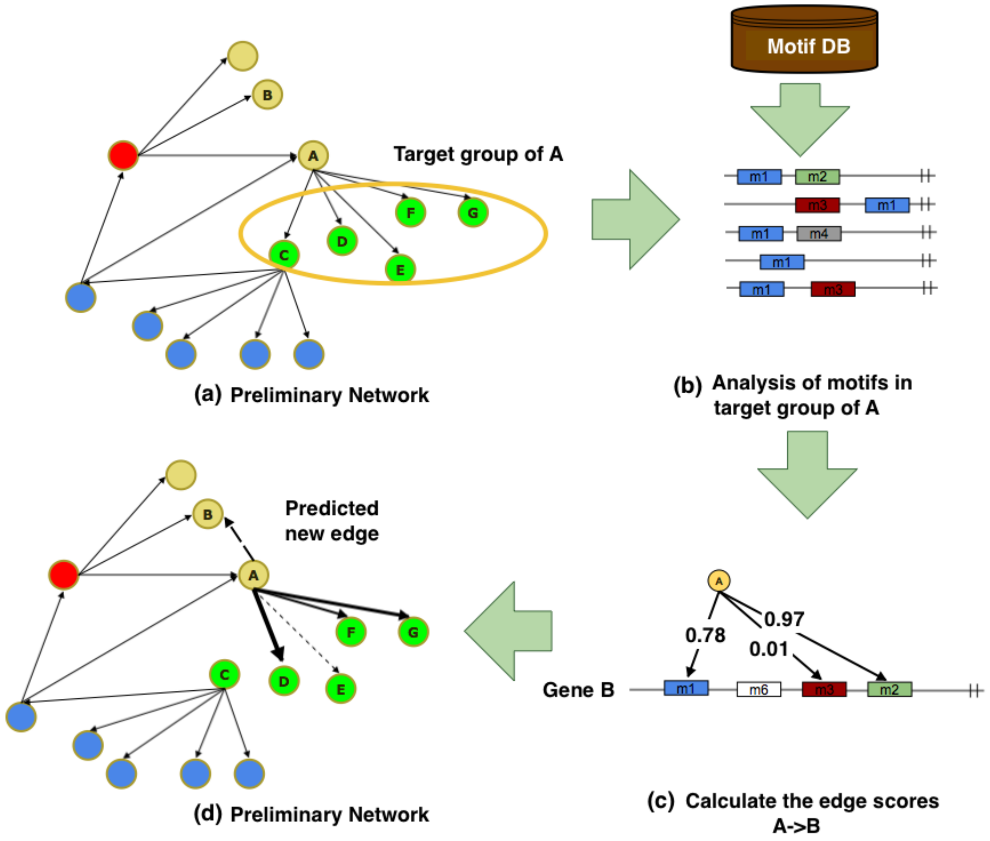


Fig. 3.2: Illustration of GRN inference using TFTS: (a) the input candidate network; (b) the database of cis-element occurrences; (c) gene-motif score; (d) refined GRN based on e^*_{ij}, where dashed edges are new proposed edges and the width of an arrow represents the confidence of the corresponding edge.

- $C_k = \{g_i | m_{ik} > 0\} \subseteq V$ denotes the set of genes that contain cis-element k at least once in their promoter region. In the illustrative example shown in Figure 3.2(b), $C_{m1} = S_A$.

For each gene motif pair (g_i, m_k) we sought to evaluate the strength of the pair using the survival function of a random variable x:

$$w_{ik} = Prob(x \geq t) = \sum_{u \geq t} p(u) = 1 - F(t) \tag{3.13}$$

where $x \sim B(n, p)$ is the binomial distribution with $n = |S_i|$ denoting the number of regulated target genes of gene i, and $p = \frac{|C_k|}{N}$ denoting the probability of observing sequence k. If the set S_i is randomly drawn from V, then w_{ik} will be large; low values of w_{ik} indicate the that cis-element k is important with respect to transcription factor i.

- **Edge scores (gene-gene relation):** The outcome from the previous process can be viewed as a set of scores of gene-motif relationships. However our goal is to evaluate the strength of edges e_{ij} between genes. For any pair of genes (i, j), we use the following value to quantify the gene-gene relation:

$$e^*_{ij} = \sum_{k \in K_j} m_{jk} (-\log w_{ik})^{\alpha} \tag{3.14}$$

where $K_j = \{k | m_{jk} > 0\}$ is the set of cis-elements identified in promoter region of gene j and α is the tuning parameter for weighing the importance of redundancy and affinity. As discussed in the previous section, the number of occurrences (m_{jk}) of cis-element k on target gene j, as well as the gene-motif score w_{ik}, are important in determining the regulation between gene i and j.

We adopt Equation 3.14 to balance between the two factors "redundancy" and "specificity" in motif analysis, because $(-\log w_{ik})^{\alpha}$ term quantifies the specificity of motif k, and m_{jk} is indicate redundancy. By increasing the parameter α, we can focus on high specificity motifs that only contained in target group S_i. By contrast, lower α values focus more on the preference of biological system to certain motif k. Furthermore, we take the sum over all motifs k so that both simple motifs and structural motifs are considered.

In our experiments, we used the value $\alpha = 0.5$. As illustrated in Figure 3.2, TFTS takes the inputs of the preliminary network and the cis-element distribution, and updates the network by removing edges with $e^*_{ij} < \tau$, where τ is a significance threshold. In our simulations, we selected different values of τ to study the performance of refinement in terms of precision.

3.5 Experiments and Results

In this section, we discuss the experiments evaluating TFTS in various datasets. We present the performance of TFTS in differentiating strong and weak edges; then we test its performance in the prediction of new edges, given a preliminary network. Finally, we use TFTS as a post-processing unit of ARACNe, and compare the performance with original ARACNe.

3.5.1 AGRIS database

In our experiments, we first used an existing GRN for Arabidopsis thaliana from AGRIS database [24]:

1. The cis-element occurrence matrix $M = m_{ik}$ is obtained from AtcisDB from AGRIS.
2. The regulatory network $G =< V, E >$ is named as AtRegNet in AGRIS, where $|V| = 8154$ and $|E| = 11356$.

• **Confirmed v.s. Unconfirmed Edges:** The edges in the database are classified into two different categories: unconfirmed and confirmed. The unconfirmed edges were obtained from text mining of literature and other computational approaches whereas the confirmed edges were validated using biological experimental approaches. We investigated whether

TFTS assigns higher scores to the confirmed edges than the unconfirmed edges. Towards this goal, we compute the weights for the existing edges in each target group and perform a t-test to compare the mean scores for the confirmed and unconfirmed edges:

$$t = (\mu_c - \mu_u) / \sqrt{\frac{s_c^2}{n_c} + \frac{s_u^2}{n_u}}, \tag{3.15}$$

where μ_c, μ_u are the mean values, s_c, s_u are standard deviations of assigned scores e^* to confirmed and unconfirmed edges respectively. n_c, n_u are numbers of confirmed and unconfirmed edges respectively. We expect TFTS to assign higher scores to confirmed edges compared with unconfirmed edges, but in the learning phase no differentiation is made between the two types of edges. Thus, TFTS calculates the score e_{ij}^* without any knowledge of the edge confidence score. As discussed in Section

In order to calculate the t-value, we focus on target sets containing at least 4 edges of each kind. As shown in Table 3.1, we summarize the mean and standard deviation assigned to confirmed and unconfirmed edges from regulator genes (i.e., the outgoing edges from influential gene are labeled as "confirmed" and "unconfirmed"). TFTS is successful in differentiating confirmed and unconfirmed edges by assigning significantly higher scores to confirmed edges compared with ones assigned to unconfirmed edges.

- **Prediction of New Edges** To assess the ability of TFTS to predict new edges we focused on the largest target group S_i where $i = \arg\max_u |S_u|$. The regulator is AT1G24260. Using TFTS we assign weights to absent edges ($\bar{E} = \{e_{ij} \notin E\}$) using gene-gene scores (shown in Equation 3.14). Since experimental methods for validation are not accessible, we used AthaMap [15] which is an independent approach using published binding sites consensus analysis to map gene-gene interactions to evaluate the performance. We proposed the top 10 target genes with highest TFTS scores, where 8 genes (as shown in Table 3.2) are also predicted to be targets of AT1G24260 by AthaMap, which implies that our prediction

Regulator Gene	μ_c	s_c	μ_u	s_u	n_c	n_u	$n_c + n_u$	t
AT1G32640	0.091	0.051	0.112	0.036	2	2	4	0.577
AT5G15840	0.126	0.079	0.209	0.090	2	3	5	0.898
AT4G18960	0.093	0.040	0.069	0.078	9	30	39	1.444
AT4G23810	0.154	0.073	0.138	0.114	12	49	61	1.950
AT3G47640	0.482	0.181	0.439	0.176	3	66	69	3.548
AT2G02820	0.430	0.095	0.400	0.200	13	230	243	0.741
AT1G14350	0.430	0.095	0.400	0.200	13	230	243	0.741
AT5G11260	0.591	0.158	0.500	0.240	221	39	260	3.050
AT3G27920	0.007	0.018	0.001	0.008	23	527	550	3.841
AT5G41315	0.037	0.060	0.044	0.081	24	693	717	3.627
AT2G20180	0.798	0.140	0.755	0.233	189	560	749	4.941
AT5G13790	0.842	0.078	0.756	0.250	22	3920	3942	3.685
AT1G24260	0.957	0.038	0.926	0.222	15	4085	4100	3.917

Table 3.1: t-test of Confirmed and unconfirmed edges.TFTS assigns significantly higher scores for confirmed edges compared with unconfirmed edges especially when sample size is big enough ($n_c + n_u > 30$)

(based on candidate network and cis-element occurrences) highly agrees with AthaMap[†].

Gene	# of matches	Average Score
AT1G14330	6 (3+,3-)	5.34
AT2G42360	2 (1+,1-)	4.59
AT3G22830	1 (1+)	6.08
AT3G61430	3 (1+,2-)	5.69
AT4G31390	1 (1-)	4.74
AT4G33800	7 (3+,4-)	5.41
AT5G05250	5 (2+,3-)	6.15
AT5G59220	4 (1+,3-)	6.22
AT1G01183	NA	NA
AT1G01250	NA	NA

Table 3.2: List of predicted new targets for gene AT1G24260 that agrees with AthaMap. # of matches refer to the number of matched fragments in the promoter region by AthaMap, where + represents the forward strand and - represents reverse strand. Average score is the confidence level of matching according to AthaMap. The details of scoring metrics in AthaMap can be found in [15]

[†] The genes in Table 3.2 is sorted by the descending order of $e*$.

3.5.2 ARACNe v.s. ARACNe + TFTS

In this experiment, we extend ARACNe with the refinement process using TFTS. By comparing the performance of GRN inference using ARACNe only and ARACNe + TFTS where the inferred network from ARACNe is used the candidate network, we found that TFTS can improve the precision of network reconstruction:

$$\text{precision} = \frac{|E^* \cap E^V|}{|E^*|},$$

where $E^* = \{(i,j)|e^*_{ij} \geq \tau\}$ is refined edge set using TFTS, and E^V denotes the experimentally verified edges (ground truth). However, the precision measure only focuses on the proposed edges from TFTS, so a very small E^* might have a very high precision. To ensure a fair comparison, we select the threshold τ to make sure $|E^*| \approx |E^V|$. The process is illustrated in Figure 3.3.

We used two data sets in this experiment:

- **Human B-cells MYC genes targets prediction:** In this experiment the candidate gene network is obtained from the steady-state expression profiles using ARACNe. Experimentally confirmed binding sites are gathered from motifMap [45, 23]. The preliminary network G is a star network and contains 2063 target genes of MYC gene. To generate the cis-element occurrence matrix M required by TFTS, we use RSA-tool [64] to annotate upstream promoter sequences of every gene.

The precision of ARACNe is 0.199 (412 out of 2063 predicted targets are true targets, i.e., $|E^* \cap E^V| = 412$) and in ARACNe + TFTS, we managed to improve the precision to 0.21.

- **Yeast regulation network in cell life cycle at α stage** We collected the gene expression data from [83] and the ground truth GRN from [1]. By using the default parameters, ARACNe inferred a very poor GRN with precision of 0.089, using TFTS as an additional refining stage, it improves it to 0.52.

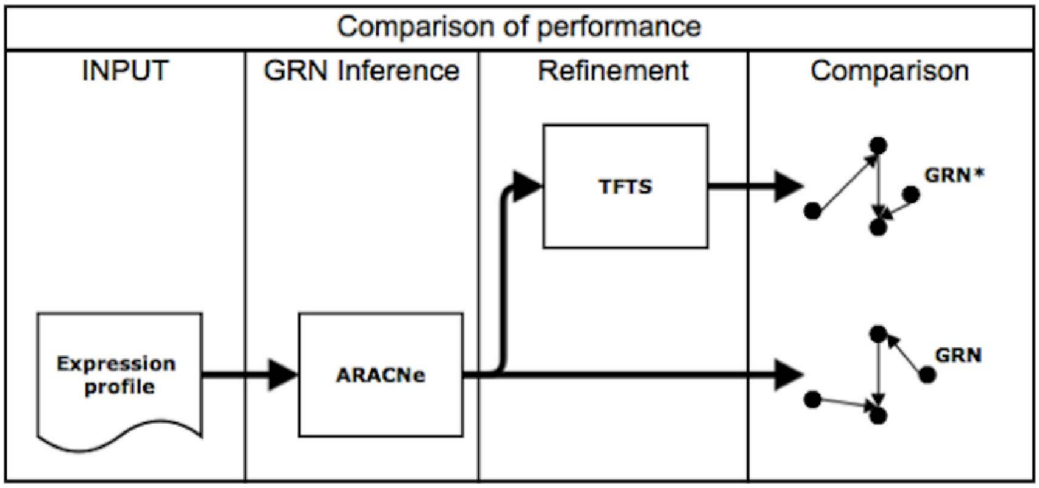


Fig. 3.3: The process of evaluating GRN inference accuracy between ARACNe and ARACNe+TFTS

3.6 Concluding Remarks

Our framework provides a new approach which takes advantage of the binding site discovery algorithms to provide a high accuracy scoring system. TFTS improves the performance of preliminary ARACNe networks by using motif information. Since we assign weights to the edges of a GRN, it is possible to assess the strength of an edge for further evaluation, such as biological experimentation. Using the weight assignment, we propose new edges. Because TFTS calculate the gene-motif scores based on the topology of the preliminary network, the performance is highly dependent on the accuracy of the network inference algorithms and on the size of target group. In the current set of experiments we have limited our investigation to simple motifs only, i.e., we have ignored the fact that multiple transcription factors, simple and structured, collaborate together in order to regulate the target genes. Further improvements can be achieved using "better" motifs. In higher-level organisms the motif information is to be extracted from the coding as well as non-coding regions. In addition, it would be reasonable to investigate if TFTS could be applied iteratively.

CHAPTER 4

FUNCTION SPECIFIC HISTONE MODIFICATION PATTERN RECOGNITION

In this work, we describe the problem of relating gene functions to histone modification combinations, and propose an innovative computational method for "n:1" pattern recognition. In Section 4.1, we introduce the background of the problem and related works. In Section 4.2, we describes the proposed algorithms, *Histone Profiling by Significance Score* (HiPSiS). In Section 4.4, we evaluate the proposed method.

4.1 Introduction

Feeding the world's population requires designing robust crops, with improved yield and enhanced resistance to diseases. In plants, histone modifications have been associated with many biological processes, including development [10], flowering-time [42], and pathogen defense [9]. To understand plant fitness, we need to understand how the histone modifications regulate development, flowering-time and pathogen defense, but only a few studies have explored the same. In this study, we attempt to understand the relationships between certain combinatorial patterns of histone methylation and acetylation in regulating plant development, flowering-time, and pathogen defense in *Arabidopsis thaliana (A. thaliana).* This is a novel step in the early stages of the *epigenetics* era, and we believe that there is a tremendous potential for the use of similar computational methods to predict how these patterns regulate plant development, flowering-time, stress and other defense responses.

Gene expression in eukaryotes is regulated at several levels, including transcription, post-transcription, translation, and post-translation. In higher organisms, genomic DNA is packaged with the help of histone proteins such as H3, H4, H2A, and H2B. Each unit of DNA and histone proteins assembly is known as the *nucleosome*. *N-terminal tails* of the histone proteins are subjected to various modifications such as acetylation, methylation, ubiquitination, and sumoylation, which regulates open/closed state of the chromatin.

Histone modification is a post-translational mechanism, which allows eukaryotes to have an additional important layer of gene regulation, by opening up the space within neighboring nucleosomes or packaging them more tightly. Relaxed nucleosomes allow access to the transcription factors, hence facilitating gene activation, whereas condensed nucleosomes restrict the access of transcription factors, resulting in gene repression. These changes may be transient or can be inherited into future generations, possibly affecting the fitness of an organism in response to various environmental stimuli.

Histone modifications such as acetylation and methylation have been shown to regulate development of plants, and recent work has shown that they also regulate stress tolerance in plants [48, 26]. Histone acetylation is mainly associated with gene activation, whereas methylation is associated with both gene activation and repression. Gene expression can be turned on and off based on the presence of active or repressive methylation marks on genes; in A. thaliana, these marks occur mostly on lysine (K) and/or and arginine (R) residues of H3 and H4 histone proteins. H3K4me, H3K36me marks are associated with gene activation; however, H3K9me, H3K27me are associated with gene repression.

Histone acetylation and methylation marks occur in various combinatorial patterns in the promoters and/or in the coding sequences of genes, leading to different outcomes of gene expression. These combinatorial patterns of the histone marks may function cooperatively or antagonistically to regulate gene expression, and have been studied mainly in humans [92], and to a much smaller extent in A. thaliana [14, 18].

This chapter reports the results of our analysis of the Chromatin ImmunoPrecipitation sequencing (ChIP-seq) dataset from Luo, et al., [59], which was produced to analyze histone modifications patterns for the regulation of natural antisense transcripts. We selected nine abundant marks from the ChIP-seq dataset, i.e., H3K4me2/3, H3K9me2, H3K27me1/3, H3K36me2/3, H3K9ac, H3K18ac, and total H3 occupancy. We have succeeded in discovering patterns that are unique to plant development, flowering-time, stress response and pathogen defense. This study is useful to understand the regulation of gene expression related to these biological processes and might be helpful in designing better crops.

Section 4.2 presents the methodology we used, including discussion of related work. Section 4.3 describes the experimental simulations and results obtained using our approach. Concluding remarks are given in Section 4.4.

4.2 Methods

In this work, we focus on ChIP-Seq data where intervals of positions on chromosomes are associated with a signal strength score. For example, Table 4.1 shows a typical bed format for ChIP-seq data defined in UCSC genome browser online service [88].

Chromosome	Start	End	Signal
chr1	1300	1521	93
chr1	2300	2412	13
chr2	100	230	19
chr3	5700	6030	45

Table 4.1: An example of ChIP-seq data set in bed format.

The strength score is a simplified data representation of histone modification activity. The high values represent reliable and active histone modification regulation and the low values represent absence of histone modification or weak affinity. They are quantified by the number of reads (read is the unit in ChIP-seq for counting matching antibodies sequences), where each interval is associated with a value. Since interval lengths may not be uniform, the common approach [75, 52] is to use consecutive fixed-length windows across the entire genome and calculate the accumulated read counts in each segment, for simplicity. We divide the raw ChIP-seq dataset into fixed-width segments of 100bp length using bedtools [72]; in our work, we consider 20 segments in the upstream 1000bp to downstream 1000bp range* . We use the empirical values for segment sizes, and up/down stream ranges suggested by [59, 52, 78].

We determine the presence/absence of each modification using a statistical threshold, as follows:

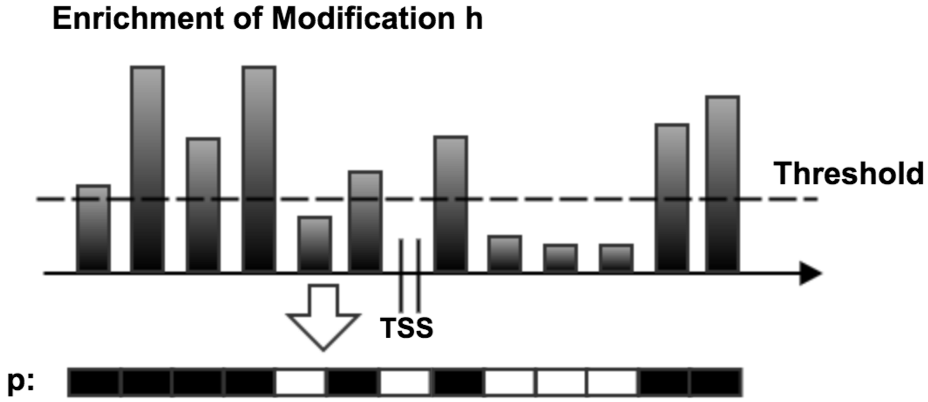


Fig. 4.1: Illustration of pre-processing of histone modification data. TSS is the starting point of transcription. h is the index of modification.

1. Let λ_j be the average enrichment level for modification j in all segments across the entire genome.

---

*The upstream and downstream regions are defined with respective the transcription starting site (TSS) and direction of coding DNA sequence (CDS). For example, the upstream and down-stream regions for gene A with TSS at 1000 and CDS=1000 → 2000 are intervals $[0, 1000]$, $[1000, 2000]$ respectively; on the other hand, the up/down stream regions of a reverse gene whose TSS=2000 and CDS= 2000 → 1000 are $[2000, 3000]$, $[1000, 2000]$, respectively.

2. Let $m_{i,j}^{(g)}$ be the accumulated read counts on the i-th fixed-width segment (of gene g) for modification j ($j = 1, 2, 3 \ldots, H$), where H is the number of histone modifications.

3. For simplicity, each modification j on gene g is represented as a binary (presence/absence) by thresholding on the highest peak:

$$x_{g,j} = \begin{cases} 1 & \text{if } \max\limits_i (m_{i,j}^{(g)} > \tau_j) \\ 0 & \text{otherwise} \end{cases},$$

   where τ_j is the threshold of histone modification j. The value for τ_j is determined by the following constraint for each modification j:

$$Prob(x_{g,j} \geq \tau_j) = e^{-\lambda_j} \sum_t^{\infty} \frac{\lambda_j^t}{t!} \leq 10^{-8}.$$

   Finally, for each gene $x_g = [x_{g,1}, x_{g,2} \ldots, x_{g,H}]^T$ is the vector representing "presence/absence" of histone modifications $1, 2, \ldots H$. The process of transformation is illustrated in Figure 4.1.

Given that the "scores" from different datasets are not identical, we adopted the genome-wide normalization method to calculate the threshold of modification enrichment in order to implement a universally applicable method. As described in step 3 above, we consider the Poisson distribution for each modification $P(\lambda_j)$. This preprocessing approach is also widely used in other histone modification pattern recognition works [30, 59, 26, 18] to focus on the impact of modifications on genes. Our choice of p-value threshold is an empirical value used by the data provider [59].

Let f denote a biological function (such as flowering-time). G_f denotes the set of genes associated with this function, and $\overline{G_f} = G \setminus G_f$ contains other genes. The ℓ^{th} pattern of histone modifications is a binary vector $P_\ell = [P_{\ell,1}, P_{\ell,2} \ldots, P_{\ell,H}]$, where $P_{\ell,j} = 1$ indicates that histone modification j occurs (otherwise $P_{\ell,j} = 0$). $X_g = [x_{g,1}, x_{g,2}, \ldots, x_{g,H}]$ is a vector that represents histone modification for gene g. We use the following notation to represent gene g contains pattern ℓ:

$$X_g \succeq P_\ell \text{ iff } \forall_j x_{g,j} \geq P_{\ell,j}.$$

- **Gene Labeling:** In reality, biological function of genes, confirmed by experimental verification, is very limited. Even in our verified flowering-time labeling [20], we cannot guarantee that $G_{\text{flowering}}$ is a complete and error-free list. So we adopt the TAIR GO term annotation dataset as the best available approximation for gene function annotation [7]. In TAIR database, the function labeling are obtained from Gene Ontology analysis via gene similarity and regulating relations. Such a labeling is neither guaranteed to be complete nor 100% accurate, but they provide a reasonable estimate for genes' possible biological functions. As a result, the preprocessed data is illustrated in Table 4.2.

Histone Modifications				Gene Functions		
m_1	m_2	$\ldots$	m_H	flowering	defense	$\ldots$
0	1	$\ldots$	1	1	1	$\ldots$
1	0	$\ldots$	1	0	*	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	1	0	1	1	*	$\ldots$

Table 4.2: The processed data with labels. Under Histone Modifications: 1 and 0 represent presence and absence of modification respectively; Under Function Labels: 1, 0 represent f and $\bar{f}$; * denote unknown functionalities

4.2.1 Related work

Although researchers have not fully understood the underlying mechanism of how histone acetylations and methylations control gene expression, several approaches have been proposed to speed up the process of biological hypothesis generation and experiment design. Subsets of genes have been used to discover significant patterns, along with genome-wide

Table 4.3: Notations and Description used in this chapter

Notation	**Description**
X_g	binary vector of histone modifications
G	Set of all genes.
$N = \|G\|$	Number of all genes
$G_f \subset G$	Genes of function f
P_ℓ	ℓ-th pattern
$G_{\ell,f}$	$\{g\|\, X_g \succeq P_\ell \,\wedge g \in G_f\}$
$\overline{G_{\ell,f}}$	$\{g\|\, X_g \succeq P_\ell \,\wedge g \in \overline{G_f}\}$
$A^{N\times L} = (a_{g,\ell})$	Indicator matrix $a_{g,\ell}$ iff $X_g \succeq P_\ell$
$n_f = \|G_f\|$; $n_{\bar{f}} = \|\overline{G_f}\|$ $n_{\ell,f} = \|G_{\ell,f}\|$; $n_{\ell,\bar{f}} = \|\overline{G_{\ell,f}}\|$	Sizes of gene sets
$r_{\ell,f} = \frac{n_{\ell,f}}{n_f}$; $r_{\ell,\bar{f}} = \frac{n_{\ell,\bar{f}}}{n_{\bar{f}}}$	Support ratios of P_ℓ in G_f and $\overline{G_f}$
$r_\ell = \|\{g\|X_g \succeq P_\ell\}\|/N$	Global support ratio of pattern P_ℓ

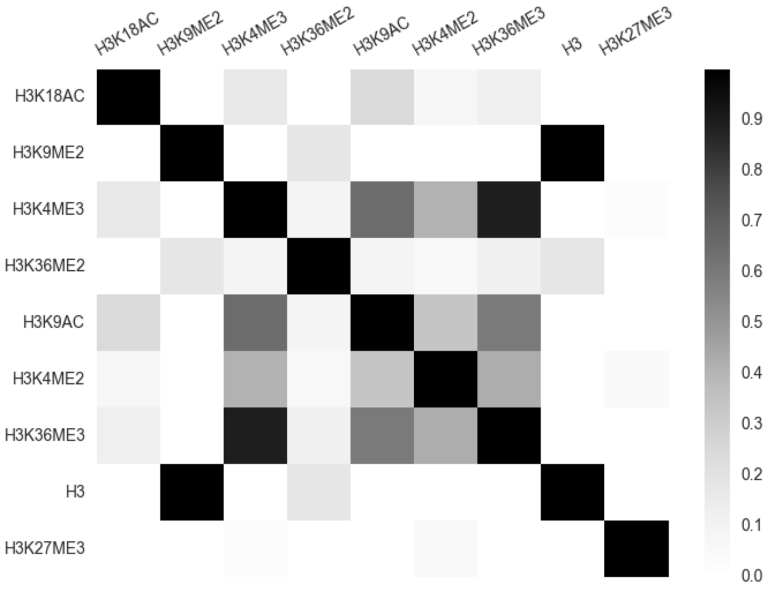
pattern discovery, as summarized below.

- **Polling of individual histone modifications** : For the set of genes G_f,

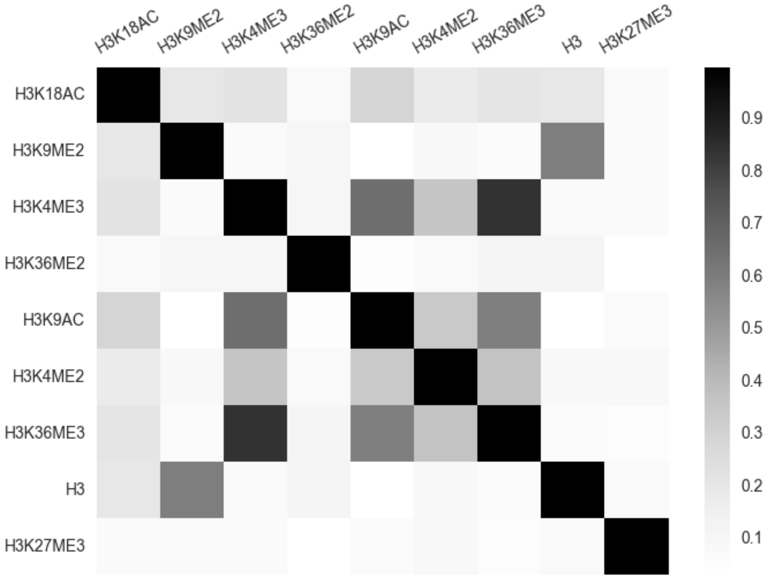
$$R^{(f)} = [r_{1,f}, r_{2,f}, \ldots, r_{H,f}]^T$$

is obtained,which summarizes the histone modifications distribution for all genes in G_f. Most works [50, 9] adopted these ratios to view distributions of different histone modifications on genes in genome wide scale. The individual histone modification ratios are not used as the final results of their analysis because of the importance of combinations rather than individual modifications has been recognized in recent epigenetic studies [67]. In this work, we use polling as our baseline against which other space transformation methods are compared.

- **Pairwise correlation analysis** : The co-occurrence strength between histone modifications j and k can be measured by the cosine similarity $\cos_{j,k} = (x_{g,j} \cdot x_{g,k})/||x_{g,j}||\,||x_{g,k}||$



(a) Cosine similarity for flowering-time related genes



(b) Cosine similarity for stress related genes

Fig. 4.2: Pairwise correlation analysis of genes labeled with f_1 = flowering and f_2 = stress.

where $g \in G_f$ (see [37, 28]). As shown in Figure 4.2, we compared pairwise similarity for two functions labels, f_1 = flowering and f_2 = stress; observing very small differences, e.g., H3 and H3K9me2 co-occur slightly more often in flowering than stress related genes.

This observation implies that pairwise correlation analysis is of limited value in determining gene functions.

- **Market basket analysis:** Frequencies (number of occurrences) of item sets have been used [58, 73, 92] to obtain combinations of important histone modifications patterns. The pattern ratio $r_{\ell,f} = n_{\ell,f}/n_f$ is used to measure the importance of pattern P_ℓ for label f, where high values are considered to indicate stronger confidence in recognized patterns. However, these recognized patterns are considered as genome-wide instead of being function specific.

- **Clustering:** The self-organizing map (SOM) approach was proposed in [67] to infer clusters of modifications, based on the raw histone modification enrichment levels $m_{i,j}^{(g)}$. But this unsupervised approach ignores the biological functions of genes, instead focusing on cluster visualization. As a result, it remains a question of how to relate different histone modification patterns to biological functions.

4.2.2 HiPSiS: Histone Profiling by Significance Score

We propose HiPSiS, an innovative method for histone modification pattern inference which focuses on evaluating patterns by evaluating a significance score. First, for each pattern P_ℓ we compute the global ratio r_ℓ, as well as $r_{\ell,f}$ for each G_f. We adopt the FP-growth algorithm for efficient enumeration and indexing [33]. Given H distinct histone modifications, 2^H patterns are possible. However, in favor of computational efficiency, we consider only the patterns that exist in the dataset. Thus, the number of combinations is bounded by min$\{N, 2^H\}$.

For each specific function f and pattern P_ℓ, we assume that $x_\ell = (r_{\ell,f} - r_{\ell,\bar{f}})/\sigma \sim N(0,1)$, where $\sigma = r_\ell(1 - r_\ell)(1/n_f + 1/n_{\bar{f}})^\dagger$. We quantify the importance of $r_{\ell,f}$ using the cumulative probability from two tails of the normal distribution, and the final score

---

[†] This statistic method is usually used in comparing mean values μ_1, μ_2 from two groups of data, which are believed drawn i.i.d from the same distribution. If the two groups truly have the same underlying distribution, then the random variable $x\ell$ should have a normal distribution.

assigned to pattern P_ℓ is $s_{\ell,f} = \log(Pr[x_\ell \leq r_{\ell,f} \cdot n_f]) - \log(Pr[x_\ell > r_{\ell,f} \cdot n_f])$. The score of a pattern is essentially the log value of odds ratios; and it represents the inclination towards a certain choice. If $s_{\ell,f} > 0$, pattern P_ℓ is considered important in function f; in contrast, patterns with negative scores are considered to interfere with function f.

In brief, the output of HiPSiS is the matrix $S = (s_{\ell,f})$ that contains the scores associated with function specific histone modification patterns. The function maps the combinatorial pattern ℓ to a real value, and quantifies its importance with respect to the function f. Higher the score, the more confident and important pattern ℓ is for biological function f. As a result, HiPSiS can be used as a function-specific pattern recognition method to select the top k patterns for each function.

4.2.3 Gene function prediction

For each function, we use a large sparse matrix $Z^{(f)}$ to represent the ownership and importance score of different patterns for each gene. The new input data is then fed into well known classification algorithms to evaluate the scoring system.

We map the original input data X on to new spaces using various pattern scoring methods to evaluate the latter. Since pairwise correlation and SOM clustering cannot quantify the importance of a particular combination of histone modifications, we performed the transformation using only the following methods.

1. Original space: We use the original binary matrix X without any modification.

2. Polling ratio weighted space: For each f-vs-rest classification problem, we transformed the original input data by: $W^{(f)} = X^{N \times H} R^{(f)}$.

3. Simple basket weighted space for function f: For each record X_g, we create a new record based on the matching results of all observed patterns. The new input data is:

$$Y^{(f)} = A[r_{1,f}, r_{2,f}, \ldots, r_{L,f}]^T \tag{4.1}$$

4. HiPSiS weighted space:

$$Z^{(f)} = A[s_{1,f}, s_{2,f}, \ldots, s_{L,f}]^T \tag{4.2}$$

In our experiments, we used the 4 different binary matrices $(X, W^{(f)}, Y^{(f)}, Z^{(f)})$ to train multiple learning algorithms.

We inferred the significant combinatorial patterns for each different biological function, and categorized genes into predicted functional groups by applying multiple classification algorithms (Logistic linear [12], Naive Bayesian [69] and Support Vector Machine with linear or Gaussian kernels [17]) to the pattern scores discussed above.

4.3 Performance of simulated data

In this section, we first evaluate the performance of HiPSiS with simulated data which is pre-populated with ground truth patterns, and compare with ChromHMM [28] in terms of pattern recognition ability. Then we evaluate HiPSiS using real histone modification data overlaid with gene ontology database as a gene function label classifier.

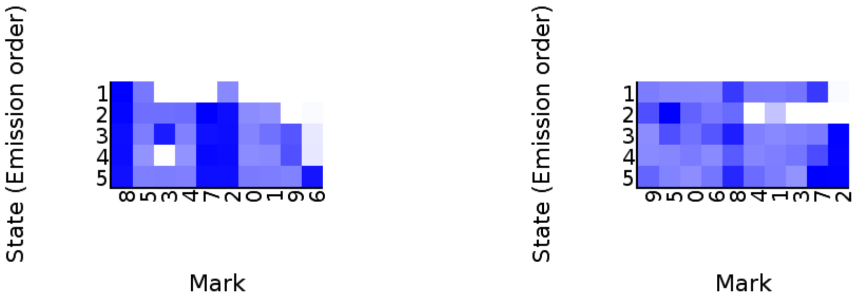
4.3.1 Pattern recognition performance

In order to evaluate the specificity and sensitivity of combinatorial pattern recognition, we compare HiPSiS with ChromHMM. We use the simulated binary data[‡] $X^{N \times M}$ where $N = 10,000$, $M = 10$, $P(X_{i,j} = 1) = 0.3$ and $P(X_{i,j} = 0) = 0.7$.

Then we plant a pattern [2, 7, 8] with probability α from a randomly selected subset G_{test}. The α parameter controls the confidence level of the planted patterns and the binary values of $X_{i,}$ are randomly toggled with probability $1 - \alpha$. Then we train the Hidden Markov Model using ChromHMM with input data G_{test}. Sequentially, we calculate the

---

[‡] In our experiments, we also tested multiple different choices for p-values.



(a) $\alpha = 0.9$ (b) $\alpha = 0.5$

Fig. 4.3: Emission probabilities in HMM model learned using ChromHMM with 5 hidden states. The $x-$axis shows elements of patterns (i.e., 0,1,2,3. . . 9) and y-axis shows the indices of hidden states of the HMM. The intensity of each square (x, y) represents the level of confidence of including x in pattern y. (a) With $\alpha = 0.9$. States 2-4 clearly captured the planted pattern [2,7,8]. (b) When $\alpha = 0.5$. The planted pattern is not obvious anymore.

pattern score using HiPSiS and find the top 5 patterns based on their scores $s_{\ell,\text{test}}$. Results show that ChromHMM is capable of capturing the pattern [2,7,8] when the confidence level is reasonably high (Figure 4.3a) but the pattern is not clear when the confidence is very low (Figure 4.3b). By contrast, HiPSiS is able to assign high scores for our planted patterns even with low confidence level. In Table 4.4, we managed to capture the planted pattern even with high noise level.

However, ChromHMM is more versatile in terms of representing patterns in which each individual modification has different probability. For example, ChromHMM is capable of capturing patterns like [0,2,7/8/9] where 7, 8 and 9 are interchangeable. HiPSiS is not suitable for recognizing such patterns because it considers [0, 2, 7], [0, 2, 8] and [0, 2, 9] as different patterns. As a result, the scores for such patterns are not significantly high.

Rank	Patterns $\alpha = 0.9$	Patterns $\alpha = 0.5$
1	*[2,7,8]*	*[2,7,8]*
2	[2,7]	[0,2,8]
3	[2,8]	[8,7]
4	[8,7]	[2,7]
5	[7]	[4,5,7]

Table 4.4: Top 5 patterns recognized by HiPSiS: results are based on the average score of repeated experiments on 20 randomly simulated datasets with the same planted pattern [2,7,8]

.

4.3.2 Real Datasets

In this section, we compare the performance of HiPSiS with methods described in section 4.2.3 for ChIP-seq dataset[§] using aerial parts of two-week-old A. thaliana [59]. This dataset contains global distribution of nine histone modifications (H3K4me2/3, H3K9me2, H3K27me1/3, H3K36me2/3, H3K9Ac, H3K18Ac, and total H3 occupancy).

Using TAIR gene ontology annotation [7], we created subsets of genes with specific functions (i.e., stress, stimulus, etc.). Luo and TAIR GO datasets[¶] are briefly described in Table 4.5.

Function	# of genes
Stress	3451
Stimulus	2938
Defense	1215
Development	2678
Flowering	212
Unlabelled	13844
Total	28523

Table 4.5: Overview of TAIR gene GO annotation dataset

[§] We accessed this dataset through the National center for Biotechnology Information (NCBI, accession number SRA010097).

[¶] The sum of the numbers of genes for each function f is not equal to the total number (28523) because some genes have multiple function annotations.

4.3.3 Verification of Predicted Candidates for Specific Functions

We assigned a score to each gene in $G_{\bar{f}}$ for each function f, using the following normalized pattern score:

$$\boldsymbol{S_f(g)} = \frac{\sum_{\boldsymbol{g \succeq p_\ell}} \boldsymbol{s_{\ell,f}}}{||\boldsymbol{X_g}||}. \tag{4.3}$$

Then, genes with high scores were selected as the potential candidates for label f. In this experiment, we only focused on flowering-time label because our domain experts created this gene list manually, whereas the labels for other functions were obtained through a keyword matching method using GO description, which is less reliable.

GeneID	Name	Verification
AT3G48430	JMJ12	flowering activator
AT3G48590	NF-YC1	ND function
AT3G63010	GID1B	activator
AT4G00650	FRIGIDA	suppressor
AT4G08920	CRY1	suppressor
AT4G15880	ESD4	suppressor
AT4G24210	SLEEPY1	activator
AT4G29830	VIP3	suppressor
AT4G34530	CIB1	activator
AT5G12840	NF-YA1	suppressor
AT5G13790	AGL15	activator
AT5G16320	FRL1	suppressor
AT5G23150	HUA1	flower development
AT5G24470	PRR5	activator
AT5G35840	PHYC	repressor

Table 4.6: Verification of 15 genes with high scores $S_{\text{flowering-time}}$, predicted to have the functionality of "flowering-time". "ND function" means that the biological function of gene NF-YC1 related to flowering-time is not determined yet

Table 4.6 shows that 14 out of 15 predicted strong candidates from $\overline{G_f}$ were verified to be correct by domain experts using an independent data source, suggesting the effectiveness of the HiPSiS pattern inference approach. We verified the roles of candidate genes by manually checking `http://www.arabidopsis.org/` database records which is independent from the GO terms used in the labeling step for detailed descriptions.[‖] These records were further validated using expression profile based experimental methods described in [20]. Whether a gene is expressed highly (in the corresponding biological process/function) is used as the ground truth to evaluate the predicted new labels for genes. Each verified gene in Table 4.6 was either confirmed experimentally in the literature, or validated with GenBank database.

4.3.4 Correlation between pattern scores and global ratios

Our main objective is to find significantly important combinations of histone modifications for each function f. The scores of function-specific patterns should be distinguishable from pattern global ratios r_ℓ. Using Equation 4.3 , each pattern is assigned with a score vector $[s_{1,f}, s_{2,f}, \ldots, s_{L,f}]^T$ to indicate the strength of relationship between patterns and label f, where L is the total number of patterns.

If a high correlation exists between a pattern score and the global ratio, then the scoring system should be considered weak because pattern importance is in accordance with global pattern score. On the other hand, if the correlation is low, then the scoring system can be considered to be informative. We compared HiPSiS with simple basket analysis because both methods evaluate the importance of combinatorial patterns of modifications. Results in Table 4.7 show that HiPSiS outperforms basket analysis because the correlation between $s_{\ell,f}$ and r_ℓ is much lower than the correlation between $r_{\ell,f}$ and r_ℓ.

Tabel 4.7 shows that the correlation values indicate the similarity between patterns from entire dataset versus specific subset (functions) of genes. In this table, we have shown that HiPSiS can distinguish patterns from a global dataset and a function specific dataset.

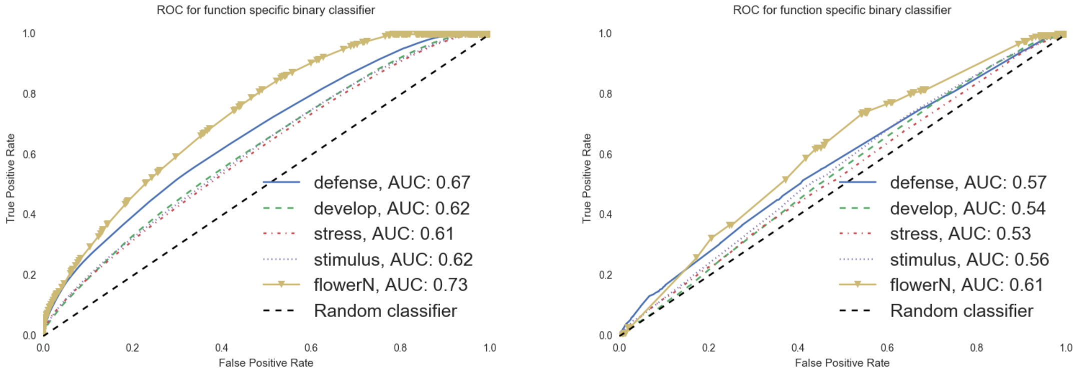
[‖] Descriptions come from definition lines supplied with TIGR gene annotation records (description is generally based on sequence similarity), as well as definition lines from GenBank records (written by the submitter). Other descriptions may be written by a curator and based upon information obtained from the available literature.

Function	HiPSiS	Market Basket
Stress	0.11	0.99
Stimulus	0.36	0.99
Defense	0.38	0.98
Development	0.08	0.98
Flowering	0.61	0.99

Table 4.7: The correlation of scores obtained using HiPSiS and simple basket analysis with background ratios.

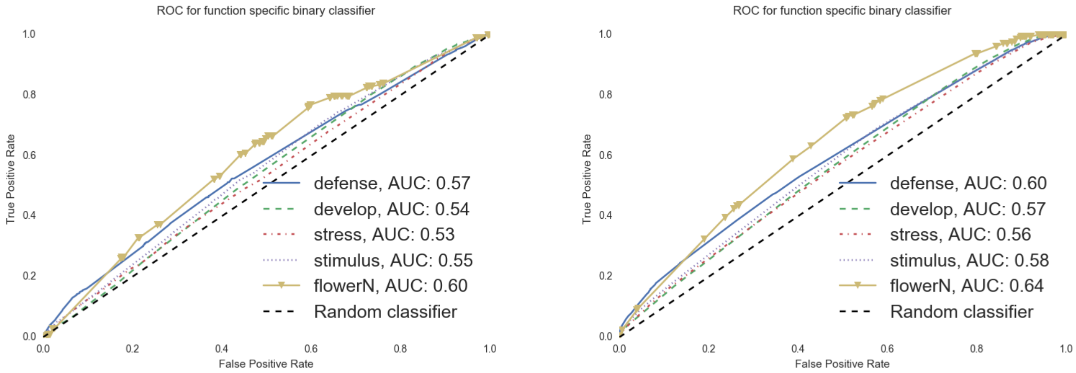
4.3.5 Function-specific patterns predicted by HiPSiS

For each function f, we applied HiPSiS to quantify the strength of each combinatorial pattern p with respect to f. Patterns with high scores are proposed as f-specific histone modification patterns, whereas others are considered to be irrelevant for function f. The proposed patterns of interest are the top five and bottom five patterns, respectively. We observed that although the histone modification H3K9ac is considered strong in multiple functions, it collaborates with different modifications in different functions. For example, H3K4me3 and H3K36me3 are the most important collaborators in "stimulus" label, whereas H3K18ac is the most important collaborator for defenses.



(a) HiPSiS $Z^{(f)}$ (b) Original Dataset X

Fig. 4.4: Classification Performance of HiPSiS and Simple Basket scoring for patterns



(a) Polling ratio weighted space $W^{(f)}$ (b) Simple Basket $Y^{(f)}$

Fig. 4.5: Classification Performance of ratios and simple basket transformed input data

4.3.6 Evaluation of HiPSiS by gene function classification

We evaluated different pattern scoring systems by projecting the original binary histone modification data X on different feature spaces. We performed a stratified 5-fold cross validation for testing with 10 repeated randomly shuffled sequences of input data X. For each function f, we evaluated the performance of Logistic Regression Classifier with transformed binary data; we also experimented with Naive Bayesian and SVM (with linear kernel as well as Gaussian kernel) classifiers, obtaining similar results.

We use the mean AUC of ROC curves to evaluate the classification performance for each function f. Figure 4.4a shows that HiPSiS performs better than other pattern scoring scores. It is noticeable that most of the binary classifiers perform almost the same as a random classifier. The main reason is the overlap between G_f and $G_{\bar{f}}$ in binary feature space. The second cause stems from the nature of function labels: we adopted the GO annotation as the function label, but these are known to be incomplete.

4.3.7 Quality of Function Labels and Performance of HiPSiS

The combinations of histone modifications are believed to contribute to different functions in biological processes. In our previous evaluations, we observe that HiPSiS perform the best on retrieving genes labeled with "flowering-time" because this label is curated using gene expression study during flowering-time of the Arabidopsis Thaliana, whereas the remaining labels are based on GO term database [56]. We evaluate the performance of HiPSiS in two more additional labels "heat" and "salt stress", which are collected using both textual and gene expression levels. The result is shown in Figure A.1

4.4 Concluding Remarks

Histone modifications play an important role in gene regulation. In this chapter, we proposed an approach to predict combinations of histone modifications that are most relevant to each biological function. We proposed a new pattern scoring method (HiPSiS), which evaluates the importance of each combinatorial pattern of histone modifications by comparing with the background ratio. Compared with other pattern scores proposed in previous work, HiPSiS was shown to be capable of inferring significant patterns which were verified by independent gene function data. We also examined the combination of different pattern scoring methods with well-known classifier algorithms to predict gene functions, and observed that HiPSiS performed the best. As an exploratory study, we list the most significant patterns for each function in Table A.3.

We were able to predict new function-specific histone modification patterns, which need to be experimentally verified in future studies. Future studies should also consider the locations and distributions of histone modifications across gene segments, which may be relevant, as implied in [37]. Additionally, we can directly examine raw enrichment values, and incorporate the locations of modifications into the feature set.

In this chapter, we discussed "n:1" patterns recognition from two universes:

- The universe of histone modifications: Ω_{mods}.
- The universe of functional labels: Ω_{labels}.

We study the associative patterns $A \in \mathcal{P}(\Omega_{\text{mods}}) \times \Omega_{\text{labels}}$, where $\mathcal{P}(\Omega_{\text{mods}})$ denotes the power set of universe Ω_{mods}. Histone modifiers are believed to collaborate with each other to maintain the normal expression of every gene in response to both internal and external stimuli to a biological system [73]. However, it still remains challenging to understand the fundamental principles of epigenetic[**] mechanisms.

[**] epigenetics is the ensemble of studies of how histone modification, acetylation and DNA methylation affect the access of chromatins.

CHAPTER 5

EFFICIENT COMBINATORIAL PATTERN RECOGNITION OF SERIES DATA

In this chapter, we discuss a new kind of histone modification patterns, as "shapes", along the span of a gene (i.e., from upstream promoter region to downstream coding region). This enables us to consider both the location (with respect to gene TSS positions) and magnitude of modification of different histone modifications. The combination of shapes (instead of presence/absence) is considered to be an informative histone pattern.

This chapter is organized as follows: in Section 5.1, we explain the motivation of shape-matching based histone modification comparison; in Section 5.2, we discuss various different series methods; in Section 5.3, we explain the three-step process of combinatorial patterns recognition of histone modification profiles; in Section 5.4, we evaluate the performance of the proposed approach against dynamic time warping and clustering as the benchmark; in Section 5.6, we summarize our findings on real histone modification data.

5.1 Introduction

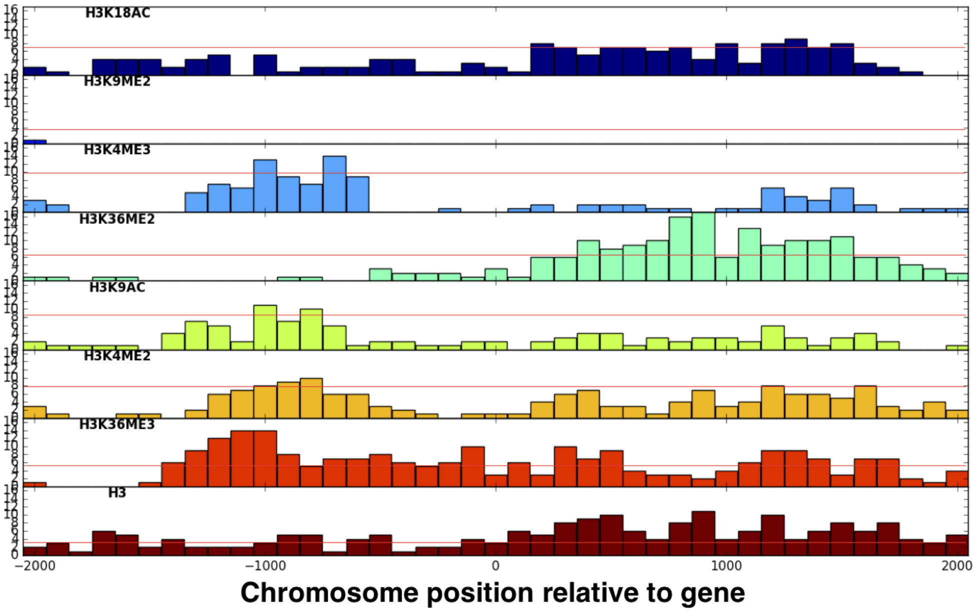
Information encoded in DNA is regulated by transcription-level regulators such as cis-elements and also by epigenetic-level components like histone modification or DNA methylation. Epigenetic regulation also controls the expression of genes, and consequently, organism phenotypes are also under the regulation of epigenetic components.

It remains challenging to understand the fundamental mechanisms of how epigenetic components regulate and control gene expression. Multiple genome-wide *in silico* methods have been proposed to search for significant "patterns" of histone modification in the recent decades. From the very beginning of epigenetic studies, researchers have hypothesized that it is the combinations of histone modifications (hypothesis of histone Code [41, 85]) that regulate the gene expression and other biological processes.

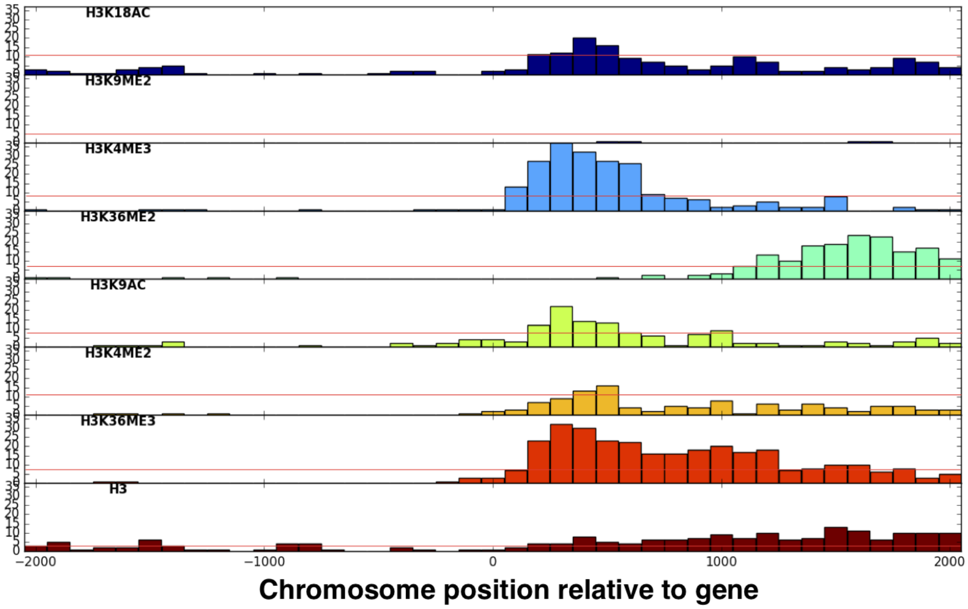
The early pioneering work [79] studied the abundance consensus of each modification individually. With the advent of highly efficient technologies like ChipSeq, recent approaches studied pairwise patterns [37, 28] and combinatorial presence/absence patterns in [58, 20]. In our work described in Chapter 4, we focused on discovering significant histone combinatorial patterns (presence or absence of modifications) in genes of specific function, and associated histone modification patterns with biological function annotations. In this chapter, we propose an innovative perspective of histone modification patterns and an efficient approach to quick indexing and comparison.

5.2 Previous work

In Chapter 4, we discussed the use of a binary vector to represent the histone modifications for each gene, and then applied simple basket analysis to enumerate and record the corresponding frequency for each combinatorial pattern. By comparing pattern frequency distributions of gene groups (each group is annotated with a biological function), we learned the significant combinatorial patterns for each function. However, localization and amplitude information of histone modifications are lost in discretization.



(a) CO



(b) Actin

Fig. 5.1: Each plot shows the raw enrichment level data for each modification for two genes: (a) CO and (b) Actin. The x-axis is the position of modification relative to TSS of each gene. The red lines in each subplots shows the thresholds using p-value 10^{-8}

Since we only considered the highest enrichment signal of each gene in thresholding process, genes with totally different enrichment levels might be converted to identical binary representations. Figure 5.1 illustrates that the thresholding of two different modification profiles and associated thresholds. Recall that binary representations, as described in Figure 4.1, $x_{gj} = 1^{\dagger}$ if we observe at least one modification level greater than threshold. For example, the binary representation of gene CO and Actin will be represented by the identical binary vector $[1, 0, 1, 1, 1, 1, 1, 1]$. On the other hand, in Figure 5.1, the raw histone data of CO and Actin, the shapes of the modifications are distinct: in CO, modification H3K9AC, H3K4ME2 and H3 show a symmetric "valley" shape, but none of modifications on Actin show such pattern. Over simplification of the previous chapter results in severe loss of information.

Given a sequence representation of a histone modification, the next concern is: how to use this information to determine a gene's functionality. In the following sections we explore this concern. Some possible similarity measures are described in the following section.

5.3 Pattern of Histone Curves

Figure 5.1 illustrates that the sequence of histone modification can be of multiple shapes. In this section, we explore if these shapes can be organized in small number of categories. Histone modification shape is defined as the shape of enrichment level curve from upstream to downstream (as shown in Figure 5.1). In this curve, interest is in the locations of high and low enrichment levels and their heights. In [58], the authors studied the histone modification shape for H3K9ac, H3K23ac and H4k12ac in human CD4-T cell dataset, and showcased the potential clusters of different histone modification shapes for genes by plotting similar histone modification curves. However, the question of how to systematically discover and use the combinations of shapes in genome wide analysis remains unsolved. In

---

[†] j is the index of histone modifiers. i.e., 0:H3K18AC, 1:H3K9ME2, 2:H3KME3. etc.

this chapter, we seek to discover the histone modification code with respect to enrichment level shapes. Towards to this goal, first we consider some similarity measure described below.

5.3.1 Similarity measures for time series

First, we discuss the naive direct comparison of series-like data points and discuss associated advantages and the disadvantage.

- **Direct comparison:** Directly calculating distance between two series (histone modification enrichment profiles) is a naive method used in many applications.

$$D(x, y) = (\Sigma_j ||x_j - y_j||^p)^{\frac{1}{p}}$$

where x, y are two enrichment curves; $p = 1, 2$ are mostly used. In this method, we compare two enrichment curves at each specific position. This method results in high dissimilarity when two time series are very similar but shifted in time with respect to each other, as shown by example in Figure 5.2. Since the specific position of each modification is not critically important in differentiating gene functions because the actual binding sites of *cis* elements vary from gene to gene [98], this measure may not be suitable for our application; and is not implemented in algorithms described below.

- **Dynamic Time Warping (DTW):** Dynamic time warping (defined in Algorithm 3) is a dynamic programming algorithm (illustrated in Figure 5.3), and is used in comparing two temporal sequences that may be shifted and vary in speed. In the context of histone modification enrichment curves, peaks observed in close proximity should be considered similar [98]. For each observation at i, DTW avoids the aforementioned problem with direct comparison by searching the neighborhood $[i - w, i + w]$ of each value for the best alignment of two series, .

However, DTW is susceptible to noise in signals, which may greatly affect the similarity

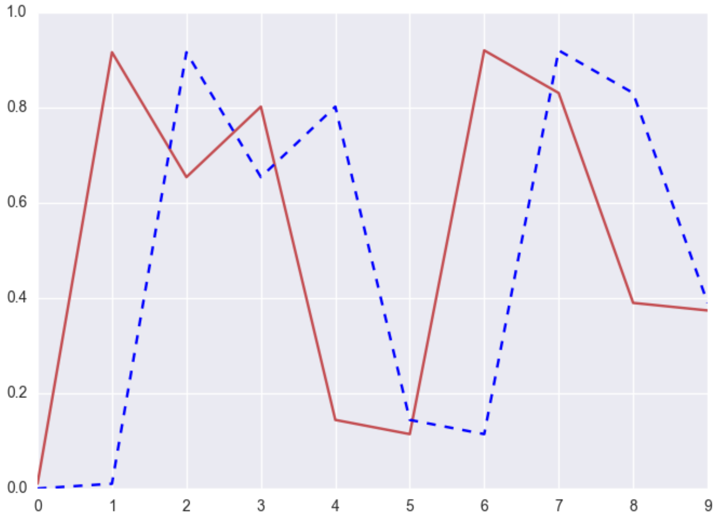


Fig. 5.2: Illustration of shifted series data and original data.

Algorithm 3 The Dynamic Time Warping algorithm with window constraint

```
function DTW($x$: array of size $n$, $y$ array of size $m$, $w$: int)
DTW ← array$[n+1][m+1]$
    for $i := 0 \ldots n$ do
        for $j := 0 \ldots m$ do
            DTW$[i][j] \leftarrow \infty$
    DTW$[0][0] = 0$
    for $i := 1 \ldots n$ do
        for $j := \max(1, i-w) \ldots \min(m, i+w)$ do
            cost← $d(x[i], y[j])$
            DTW$[i][j] = cost + \min \begin{cases} \text{DTW}[i-1][j-1], \\ \text{DTW}[i][j-1], \\ \text{DTW}[i-1][j] \end{cases}$
    return DTW$[n][m]$
```

measure between enrichment curves [70]. Another problem with DTW is the high time complexity which is $O(n^2)$, where n is the length of series data. As a result the overall time complexity becomes $O(N^2 n^2)$, where N is the total number of genes.

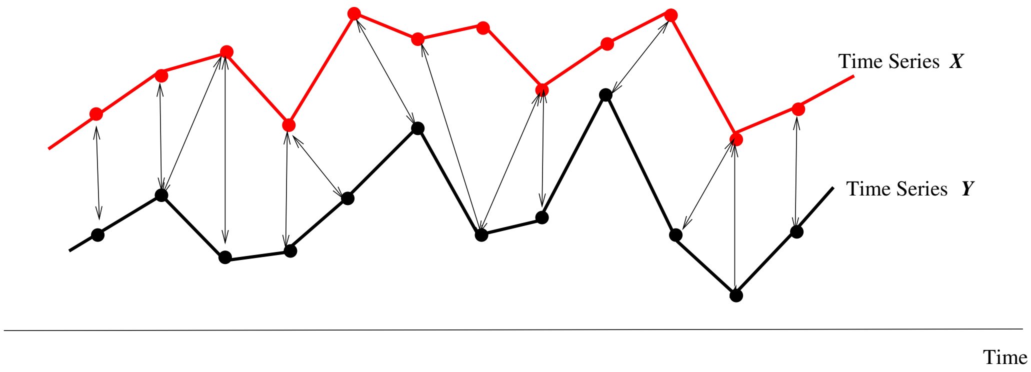


Fig. 5.3: Illustration of dynamic time warping method of time series comparison. The arrows represent the optimal alignment between sequences.

5.3.2 Algorithm for shape based pattern recognition

In this section, we introduce a three-step indexing and clustering method for combination pattern recognition.

1. **Approximation:** Shortening the series using approximation methods such as: piecewise aggregate approximation or discrete wavelet transformation;
2. **Discretization:** Discretizing of shortened series using symbolic aggregation;
3. **Clustering:** Clustering of transformed series using affinity propagation. The centroids of every cluster are used to represent the shapes for all cluster members;

Detailed discussion of these three steps is provided in the following.

1. Approximation step: For each modification h, let vector $X_h^{(g)} = (X_{h1}^{(g)}, X_{h2}^{(g)}, \ldots X_{hL}^{(g)})$ denote the raw segmented ChipSeq data on gene i, where L is the total number of windows in the proximity of gene g's TSS. We create an approximation (shortened version) $\widetilde{X}_h^{(g)} = (\widetilde{X}_{h1}^{(g)}, \widetilde{X}_{h2}^{(g)}, \ldots \widetilde{X}_{hP}^{(g)})$ for each vector $X_h^{(g)}$ using the either of the following methods:

Piecewise Aggregate Approximation (PAA) approximates a time-series of length n into a vector $\widetilde{x} = (\widetilde{x_1}, \widetilde{x_2}, \ldots, \widetilde{x_P})$ of length $P \leq L$ where each $\widetilde{x_i}$ is calculated as follows:

$$\widetilde{x_i} = \frac{P}{L} \sum_{j=a(i-1)+1}^{ai} x_j$$

where $a = \lfloor L/P \rfloor$ is the width of the segmenting window. Essentially, PAA computes the average of values within each window to summarize original data.

Discrete Wavelet Transform (DWT) is a transformation, designed to turn a series data in the time domain [*] into a sequence of coefficients with an orthogonal basis. A typical application of DWT is signal de-noising in digital signal processing. DWT is preferable to Fourier transform for our problem because enrichment curves don not show periodic behavior, and Fourier transform cannot preserve the information about the localization of each frequency, which is considered to be important.

Because of the simplicity and high performance in de-noising properties as discussed in [90] on original enrichment curve, we use the Haar wavelet[†] for high/low pass filters. The approximating process is composed of two steps defined as follows:

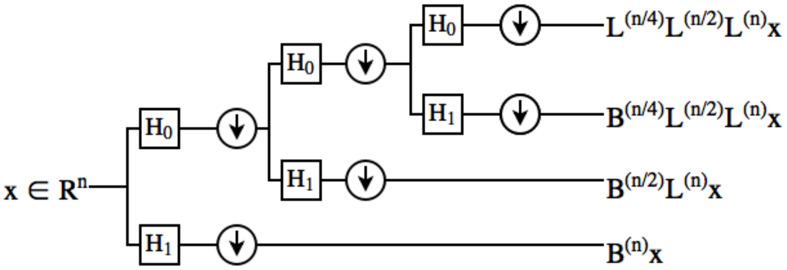


Fig. 5.4: Wavelet decomposition step with 3 levels ($k = 3$)

- **Decomposition:** The original series x is decomposed using wavelet transformation, as shown in Figure 5.4. We iteratively apply the wavelet decomposition to obtain coefficients of down-sampled data from low pass filters until no further decomposition is possible (i.e., the dimensions of down-sampled data from low pass filter is 1). Let

---

[*] In the context of histone modification, the segments of enrichment levels over locations in the gene are analogies to the time domain.

[†] We also experimented with Daubechies wavelets. Haar gave the best performance in terms of agreement with DTW and running time.

$x \in R^n$ denote the original data points[‡] series. Let

$$L^{(n)} = D^{(n)} H_0^{(n)},\ B^{(n)} = D^{(n)} H_1^{(n)}$$

denote the filter bank at each iteration of transformation, where H_0 and high H_1 are low and high pass filters. $D^{(n)}$ denotes the down sampling matrix. The result of decomposition at level n using wavelet transformation is a list of coefficients (as shown in Figure 5.4):

$$w = L^{(1)} L^{(2)} \dots L^{(n)} x,\ B^{(1)} L^{(2)} \dots L^{(n)} x,\ \dots, B^{(n)} x.$$

- **Reconstruction:** The original series x can be fully retrieved by the iteratively apply inverse transformation using coefficients obtained in decomposition step. However, in most instances an approximated $\tilde{x}$ is obtained by keeping the important traits without high frequency noise (Figure 5.5 shows an example). In this process, we use the output of the previous decomposition step to reconstruct a vector $\tilde{x}$. In Haar wavelet example, the first iteration will use coefficients $w_1 = L^{(1)} L^{(2)} \dots L^{(n)} x$ and

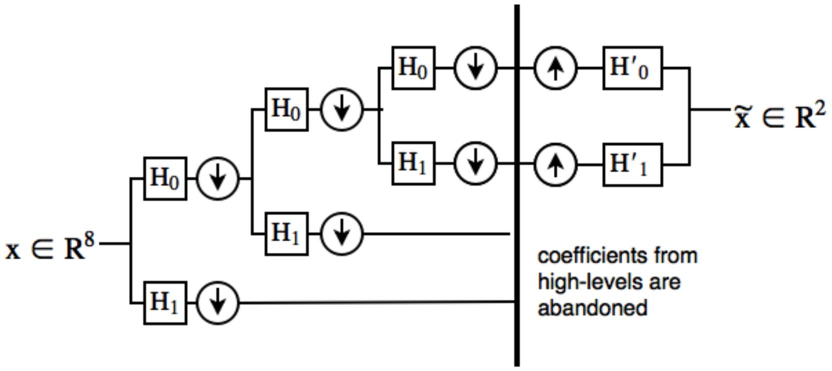

H0
H1
H0
H1
H0
H1
H'0
H'1
$\mathbf{x} \in \mathbf{R}^8$
$\tilde{\mathbf{x}} \in \mathbf{R}^2$
coefficients from high-levels are abandoned

Fig. 5.5: Reconstruction with approximation using first 2 levels, the original series of length 8 is shortened to 2 (i.e., $k = 2$).

[‡] Unlike PAA approximation, DWT approximation requires $n = 2^k$. We used constant padding (i.e., replicating border elements to make the dimension input data into a power of 2). For example, given the input data $x_1, x_2 \dots x_L$, where $L = 12$. DWT requires 2 more values on each end to execute (i.e., the length becomes 16). Constant padding will extend the input vector by replicating the values on borders. As a result, the new input for DWT is $x_1 x_1 | x_1, x_2 \dots x_n | x_n, x_n$.

$w_2 = B^{(1)}L^{(2)} \dots L^{(n)}x$ as the inputs to the reconstruction part, and then the intermediate result with two more coefficients are used for the next iteration. The details of DWT based approximation is provided in Section A.4.

After the approximation step, the original data is transformed into a shorter compact approximation.

2. Discretizing step using symbolic aggregation: SAX (Symbolic Aggregate) serves as a quick indexing method for time series data applications. Each enrichment value is represented by an alphabetical vector. The following steps are used to create such an indexing for histone modification data:

1. Z-normalization is performed at each position j of shortened histone modification data $\widetilde{X}_{hj}^{(g)}$, where j is the position, h is the modification index and g is the gene id. The z-transformation is defined as follows:

$$D_{hj}^{(g)} = \frac{\widetilde{X}_{hj}^{(g)} - \mu_{hj}}{\sigma_{hj}}, \tag{5.1}$$

where μ_{hj} and σ_{hj} are mean and standard deviation of values $\widetilde{X}_{hj}^{(g)}, g = 1, 2 \dots N$

2. Equal bandwidth discretization method is used for normalized levels D_{hj} with respect to each modification h and position j.

3. As a result, the modification enrichments for gene g are transformed into a vector of alphabets $[c_1, c_2, \dots, c_m]$. Let $[L(c_j)U(c_j)]$ represent the discretization interval of alphabet c_j at position j.

After the SAX discretization, the following pairwise distance matrix is calculated: for any pair of two SAX vectors associated with two series $A = [a_1, a_2, \dots, a_P], B = [b_1, b_2, \dots, b_P]$, the distance is defined as the following:

$$D(A, B) = \sqrt{\sum_{j=1}^{m} \min(||U(a_j) - L(b_j)||, ||U(b_j) - L(a_j)||)^2} \tag{5.2}$$

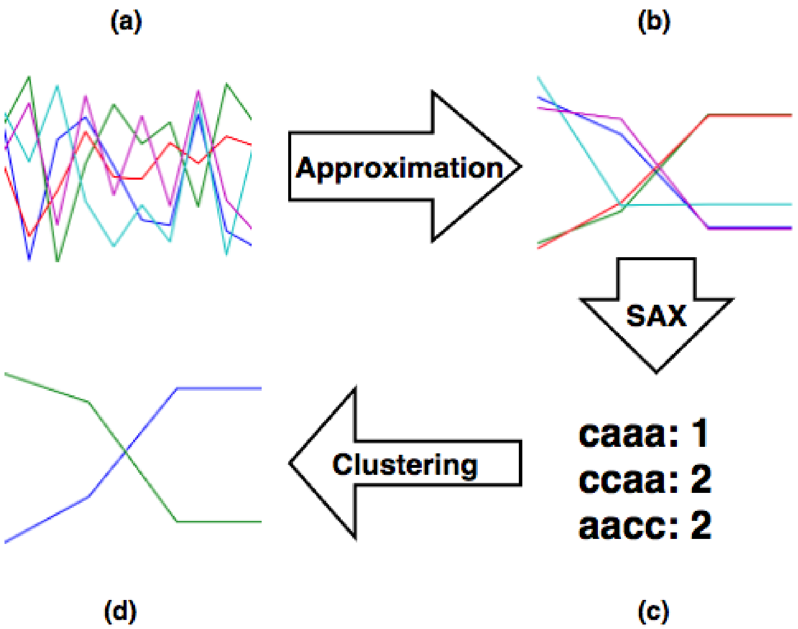


Fig. 5.6: An illustration of the proposed approach with 5 series of length 10: (a) the original series; (b) approximated and normalized data; (c) discrete representations and number of occurrences; (d) the 2 centroids yielded by clustering.

We obtain a pairwise distance matrices for each modification $h = 1, 2, 3 \ldots H$.

3. Clustering step: We apply clustering on the distance matrix calculated using Equation 5.2 for each histone modification h. Then each gene is assigned to a cluster, and we use $o_h(g)$ to denote the cluster assignment for gene g with respect to histone modification h. The cluster assignment for each gene g is summarized in vector:

$$O(g) = [o_1(g), o_2(g) \ldots o_H(g)],$$

where each $o_h(g)$ denotes the clustering assignment to gene g for histone h.

Figure 5.6 illustrates the proposed 3-step process of getting shape patterns for each modification. The cluster centroids are used to depict the shapes for the specific histone modification. In our experiments, we used both affinity propagation and k-means clustering for step 3.

5.4 Comparison of Series Similarity Measures

In the previous section, we introduced dynamic time warping and our proposed procedure. However, as realized in many studies in series data analysis researches ([13, 57, 38]), the optimality of one similarity measure is highly dependent on the applications. For example, for electromagnetic waves data, the best measure is to compare the inferred characteristics of waves (i.e., frequency, amplitude and shift); for applications like gene expression level comparison (as discussed in Chapter 3) correlation based methods are often used. In histone modification pattern analysis, the optimal choice of comparison is still open to discussion. Because of the lack of understanding of how exactly histone marks collaborate with each other in biological regulation, we cannot quantitatively evaluate the performance of each similarity measure with labels assigned by biological researchers. There are case studies regarding the value of histone modification curve shapes in [92, 79], and genes are grouped by the location where the modifications are most enriched (i.e., upstream of TSS or down stream of TSS). In this work, we propose a systematic method to calculate similarity and group genes based on their shapes.

In this section, we compare our 3-step algorithm, described in the previous section against dynamic time warping followed by the same clustering algorithm. We use dynamic time warping as the benchmark for variations of our proposed methods: (PAA + SAX or DWT + SAX). The agreement ratio is defined as:

$$\frac{2\sum_{i=1}^{N}\sum_{j>i}^{N}\mathbb{1}(\mathcal{L}_{\text{dtw}}(i,j)=\mathcal{L}^{*}(i,j))}{N(N-1)}, \tag{5.3}$$

where

$$\mathcal{L}_{\text{dtw}}(i,j) = \begin{cases} 1 & \text{if } O_{\text{dtw}}(i) = O_{\text{dtw}}(j) \\ 0 & \text{otherwise} \end{cases}$$

$$\mathcal{L}^*(i,j) = \begin{cases} 1 & \text{if } O^*(i) = O^*(j) \\ 0 & \text{otherwise} \end{cases}$$

denotes whether gene i, j are assigned to the same cluster by DTW ($\mathcal{L}_{\text{dtw}}$) and our proposed procedure $\mathcal{L}^*(i,j)$, respectively.

- **Random Series:** First, we performed the experiment using randomly generated data, where each value $X_{h,j}^{(g)}$ is drawn from a uniform (i.i.d.) distribution over interval $[0, 1]$. No inter-column or inter-row column dependencies were introduced during the process of simulation. For simplicity, we only simulated the data for one histone modification ($H = 1$), and number of genes $N = 2000$ with $L = 10$ observations as the artificial histone modification data.

	$\lvert\Sigma\rvert = 3$			$\lvert\Sigma\rvert = 4$		
	$P = 2$	$P = 4$	$P = 8$	$P = 2$	$P = 4$	$P = 8$
DWT + Affinity Prop.	0.25	0.49	0.63	0.27	0.52	0.67
PAA + Affinity Prop.	0.23	0.50	0.59	0.29	0.54	0.66
DWT + k-Means	0.22	0.56	0.71	0.27	0.60	0.73
PAA + k-Means	0.27	0.49	0.69	0.25	0.61	0.79

Table 5.1: Agreement ratio (defined in Equation 5.3) between proposed methods and DTW for simulated data. $|\Sigma|$ denotes the size of the alphabet used in the symbolic aggregation step; P denotes the length of series after approximation. These results are the mean values of 10 trials.

In this experiment, we use the randomly simulated data with uniform distributions to test the agreement levels between clustering results from distance matrices calculated using proposed approach (i.e., approximation + SAX discretization) and DTW. First, we approximate the original series into lengths $P = 2, 4, 8$); and then apply SAX using different sizes of alphabets $|\Sigma|$; followed by a clustering on the transformed series. On the other hand, DTW is used to calculate the pairwise distance matrix using Equation 5.2. Both distance matrices are provided to a clustering algorithm (affinity propagation or k-means) to get the cluster assignments for each gene $g = 1, 2, \ldots N$. Then we use Equation 5.3 to compare PAA+SAX and DWT+SAX to dynamic time warping.

As shown in Table 5.1, when both P and $|\Sigma|$ are large, the clustering assignment is similar for both the proposed procedure and DTW. It is noteworthy that $k-$means yield higher agreement ratios because the enforcement of number of clusters.

- **Real data test:** Next, we use the raw histone modification data for modifier H3 from [59] (where $N = 28000, L = 20$) and applied the same procedure to test the agreement ratio between proposed method and DTW, and observed higher agreement levels for both PAA+SAX and DWT+SAX (e.g., 0.81 when $P = 8$ and $|\Sigma| = 4$. This is because random data are not naturally separable and there are real clusters in histone modification data).

The average running time of DTW+affinity propagation is 4578.3 ms and our proposed new algorithm takes 23.4 ms on average[§]. Our proposed procedure excels in computational efficiency in shape based pattern recognition.

5.5 Histone Curve Shapes

In this section, we report the observed, patterns of shapes in each modification for Arabidopsis thaliana at young stage [92]. We applied the aforementioned procedure (i.e., DWT approximation + SAX + Affinity Propagation Clustering), and observed the patterns of shapes in each modification. We used $P = 4$ and $|\Sigma| = 4$ as the parameters.

As shown in Figure 5.7, each curve is regarded as a signature modification shape of its corresponding histone modification. In this experiment on real dataset, we observed that the recognized signature shapes of compressed data are common across multiple histone

---

[§] Experiments were conducted using a single-thread on i7-3770k CPU with 16GB memory.

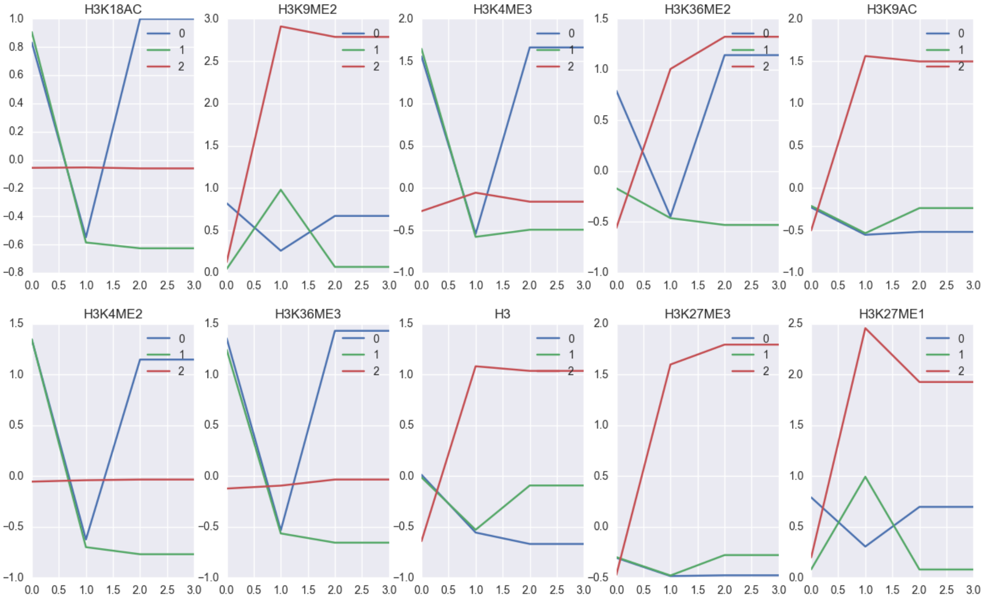


Fig. 5.7: Histone cluster centroids using approximated data ($P = 4$). Each curve represent a cluster centroid. X-axis shows the compressed indexes of original data (i.e., $j = 0, 1, 2, 3$.)

modifications. They can be summarized into the following categories:

1. $'L'$-shape: (curves labeled = "1") In Figure 5.7, histone modifications of H3K18AC, H3K4ME2, H3K36ME3 and H3K4ME3 have this shape indicating that the peak of modification occur at the far upstream of genes. The modification levels are low on the other regions of genes.

2. $'V'$-shape: (curves label = "0") In Figure 5.7, histone modifications of H3K18AC, H3K4ME2, H3K36ME3 and H3K4ME3, H3K36ME2 show a valley like symmetric modification enrichment shape. Upstream and downstream regions of genes are both modified.

3. $'\Gamma'$-shape: (curves label = "2") In Figure 5.7, histone modifications of H3K9ME2, H3K36ME2, H3K9AC, H3, H3K27ME3 and H3K27ME1 show a modification pattern where upstream is not regulated by histone modification, and downstream coding regions are highly influenced.

4. $'0'$-shape: other curves with relatively low histone modification enrichment everywhere fall into this category.

5.6 Case Study in Histone Modification Shape Patterns for flowering-time Genes

In Chapter 4, we proposed a binary version of histone code and studied its value in the prediction of gene functions, in this study we achieved the highest performance for flowering-time genes mainly because we have more confidence in associated functions; these genes are labeled by studying the expression levels of genes in flowering-time. Performance was relatively poor for the other labels (i.e., stress, defense, stimuli and development) curated from GO terms [7], which are associated with low confidence. Using pattern recognition algorithm HiPSiS, we found that the combination of histone modifiers: H3K4ME3, H3K4ME2 and H3K36ME3 are of great importance in flowering-time related genes. Inspired with this success, we investigate whether there exist subgroups of genes in these 303 manually verified flowering-time genes; particularly, are there different shape patterns of these 3 histone modifications in these genes.

As an exploratory study, we applied the aforementioned procedure to raw histone modification enrichment data of the 303 genes. In order to simplify the process, we only focus on modifications H3K4ME3, H3K4ME2 and H3K36ME3. After the proposed 3-step procedure, each gene g is associated with a categorical vector. i.e., each gene is transformed into a vector of nominal values:

$$O(g) = [o_1(g), o_2(g), o_3(g)],$$

where $o_i(g) \in \{L, V, \Gamma, 0\}$.

We study the distribution of $O(g)$ for all 303 flowering-time genes, and observed that two patterns constitute the majority: Namely LLL, VVV with ratios of 0.28 and 0.25. The ratio of the third most frequent combination, LVV, is 0.11. According to the pathway analysis from [46] by biologist experts, we found 43% of genes in the first list (with pattern LLL) participate in the light pathway, and only 3% of genes in the other list VVV participate in the light pathway. Since our proposed definition of shape-based histone modification patterns is new, so the interpretations of the recognized combinations of shapes are still open to discussion. In this case study, we showed the potential associative patterns between combinations of shapes and biological regulation pathways. In our experiment, we also adopt different parameters $P = 8, |\Sigma| = 8$. As expected, there are more number of clusters, and thus more shape patterns. In those experiment, we observe the similar results, i.e., majority of genes belong to small number of shape patterns.

The two subclasses (LLL & VVV), in flowering-time genes, are listed in Table A.1 and Table A.2 in the Appendices. In our study, we repeat the exploratory shape patterns analysis for other labels (i.e., stress, development and stimuli), and observe existence of dominating shape patterns too. Due to the lack of biological experiment at this time, no further interpretations are available at this time for these recognized patterns.

5.7 Concluding Remarks

In this work, we proposed an innovative perspective of histone code pattern by introducing the "shapes" of histone curves (i.e., where do enrichment occur with respect to TSS of genes). In addition, we use a 3-step process to compress, index and cluster histone modification series data. Furthermore, we compare the clustering results of the proposed approach to dynamic time warping and obtained similar results with significant improvement in efficiency and storage size. We use the clustering assignment vectors $O(g)$ to represent the new approach for analyzing combinations of shapes. In our case study of real histone modification data, we observed that genes from different pathways have different combinations of shapes. Our work serves as a starting point of research topics in "histone codes" by introducing the shape matching perspective rather than only considering the effect of "presence/absence" of histone modifications.

CHAPTER 6

CATWORKS: ASSOCIATIVE PATTERN RECOGNITION BETWEEN COMBINATIONS OF CATEGORICAL DATA

In this chapter, we propose the problem of $n : m$ associative patterns recognition and an algorithm to retrieve the hidden associative patterns between two universes. Unlike $1 : 1$ or $n : 1$ patterns, here we focus on combination to combination associative patterns.

In Section 6.1, we first introduce the problem of associative pattern recognition of categorical data. In Section 6.2-6.3, we explain in details the procedure of recognition. In Section 6.4, we evaluate the proposed algorithm with simulated data and real histone modification data.

6.1 Introduction

Pattern association mining is a problem that originated from our previous study in histone modification pattern analysis where we sought to learn the significant patterns (combinations of histone modifications) for a specific group of genes which are believed to participate in the same biological process, or belong to the same group [96]. However, we only focused on the synergistic behavior of different modifiers, and the combinations of different labels are ignored. For example, questions such as what are the most important histone patterns of genes labeled as "stress" and "development" are not answered. In this work, we proposed an efficient pattern association mining algorithm which is applicable to histone modification analysis, and also for generalized categorical data pattern association learning.

6.2 Problem Formulation

In this chapter, we introduce a new information retrieval (IR) algorithm, *Catworks*, to address the pattern association mining problem. In our problem formulation, we are mainly interested in scenarios where an object's features take categorical values.

6.2.1 Categorical data

A categorical variable is an ordinal variable (such as letter grades A, B, etc in a course which satisfy the order $A > B > C > F$) or takes nominal attributes (such as blood types, colors of fruits or ingredients in recipes) where there is no order between values. A categorical variable takes on a value in a fixed set.

In the problem that we have addressed in this work, both the inputs and desired outputs take categorical values as illustrated by a few examples discussed below.

When objects being studied are grouped into categories based on some qualitative trait the resulting data are merely labels or categories. In natural language processing with social media, tweets have genre attribute (such as sports, cuisine) or sentiment attribute (such as sad or uncertain). For example consider the tweet from former president Obama when he left the Oval office:

> *Thank you for everything. My last ask is the same as my first. I'm asking you to believe. Not in my ability to create change, but in yours.*

The key-word based processing results in categorical values: ("thank", "last", "believe", "change" etc.). The sentiment labels associated with this post may be "sad", "encouraging" and "grateful". In bioinformatics, we consider, for example, histone modifications of a gene; thus giving rise to a set of modifications that take place in the gene. On the other hand, we may have the knowledge of gene functionality, such as response to stress or heat.

In general, we consider different categorical attributes as different "facets" of an object. It is possible for a single data point to have multiple distinct values in each attribute. For example, in sentiment attribute of a post, it may have both "sad" and "encouraging". We seek to discover the associative patterns of combinations of categorical values between different "facets" of data. For each post, there may exist many different facets.

6.2.2 Data transformations

Let $< A_{i1}, A_{i2}, \dots, A_{iN} >$ be the i^{th} data point with L facets where $A_\ell \subseteq \Omega_\ell$ represents nominal values in categorical attribute ℓ, and $\Omega_\ell = \{\omega_{\ell 1}, \omega_{\ell 2} \dots\}$ denotes the universe of all nominal values for facet ℓ.

In the tweet post example, for instance, various keywords constitute the one universe, and the set of all sentiments becomes another universe. In this study, we limit the number of facets to $N = 2$. Furthermore, Let

$$\Omega^{(L)} = \{\omega_1^{(L)},\ \omega_2^{(L)} \dots\}, \Omega^{(R)} = \{\omega_1^{(R)}, \omega_2^{(R)} \dots\}$$

denote these two universes, and $< A_i^{(L)}, A_i^{(R)} >$ denote the i^{th} data point for convenience. For each facet we transform the nominal values to a binary vector representation as follows:

Let $d^{(L)} = |\Omega^{(L)}|$ and $d^{(R)} = |\Omega^{(R)}|$ denote the size of universes. For a nominal value based data point $< A_i^{(L)}, A_i^{(R)} >$, we transform it to a pair of binary vectors $x_i = [x_{i1}, x_{i2} \ldots, x_{id^{(L)}}], y_i = [y_{i1}, y_{i2} \ldots, y_{id^{(R)}}]$ to represent the nominal values, where

$$x_{ik} = 1 \text{ iff } \omega_k^{(L)} \in A_i^{(L)}, \text{ otherwise } 0$$

$$y_{ik} = 1 \text{ iff } \omega_k^{(R)} \in A_i^{(R)}, \text{ otherwise } 0.$$

6.2.3 Objective

Let D be the dataset consisting pairs of binary vectors (x_i, y_i) for $i = 1, \ldots m$, where x_i and y_i are input and output vectors respectively. Let $\mathcal{P} \subseteq D$ be a special subset whose elements satisfy the following constraint:

> If $(x_i : y_i) \in \mathcal{P}$, then for a given x_i the associated y_i is unique.

In other words, observations in $\mathcal{P}$ satisfy the uniqueness property:

$$x_i = x_j \text{ iff } y_i = y_j \tag{6.1}$$

These special elements are called *patterns*; $\mathcal{P}$ is considered to be the ground truth of the association. Other elements of D are not required to satisfy this constraint and consist of elements that are variations of a pattern, due to noise and other perturbations. In a noise-free scenario, i.e., when $D = \mathcal{P}$, the task of association is trivial due to exact matching. On the other hand, in a real world dataset consisting of noisy data with missing and/or wrong labels, deriving association rules is a difficult task.

Our objective is to create a retrieval procedure R which, for a new given input, produces a vector $(x_i, y_i) \in \mathcal{P}$.

Notation: $\alpha = \frac{|\mathcal{P}|}{|D|}$; i.e., α denotes the ground truth ratio.

6.2.4 Related works

In this section, we discuss the related areas of study and the difficulties of pattern association mining.

Association Rule Mining (ARM) At a first glance, the association pattern concept is very similar to association rule mining [33, 35] which is widely used in transaction data analysis to study the relations between subsets of items. The confidence of any rule $c(X \to Y) \in [0, 1]$ is essentially the conditional probability of observing subset $X \cup Y$ given that X is observed.

Although the vanilla ARM algorithm does not directly support the concepts of two categorical attributes, ARM can still be used for pattern association mining by simply removing learned rules $X \to Y$ when $X \nsubseteq \Omega^{(L)}$ or $Y \nsubseteq \Omega^{(R)}$

ARM is designed to discover strong rules of subsets of items. For example, in point-of-sale analysis in supermarkets, sales analyst are interested in rules such as {onions, potatoes} → {burger} so that they can make decisions to increase profit. However, whether the sales record contains other items is not of interest. In pattern association mining, we seek to discover the relation between combinations instead of subsets. Namely {onions, potatoes} and {onions, potatoes, tomatoes} are considered totally different combinations. For example, in gene histone modification analysis, we cannot assume genes with histone modifiers $H \supseteq H'$ will have function labels F' even if we know $H' \to F'$. Namely, the partial ordering monotonicity $X' \subseteq X \implies Y' \subseteq Y$ is not guaranteed.

Classification Approaches Classifiers are reasonable choices for pattern association mining. Binary relevance [87, 74] is a meta-algorithm for multi-label classification problems. The process trains $d^{(R)}$ number of binary classifiers independently using a "one-vs-rest" (OVR) strategy.

For each classifier j , $\{D_i^{(L)}, D_{ij}^{(R)}\}$ is used as the training data. By using $d^{(R)}$ classifiers, we can determine the final binary output vector.

The main gap between classification and pattern association problems lies in the role of "training data". In most classifiers, the objective is to minimize some cost function which reflects the model error from comparing model predicted value $\hat{y}$ and y. Whereas in the pattern association mining problem, the real patterns $\mathcal{P}$ lie buried in D. As a result, classifiers suffer from the problem of noise or useless data.

Bidirectional associative memory (BAM) is a type of recurrent neural network which is used for hetero-associative content based memory retrieval. Given a set of hetero-association patterns x_i, y_i it will store the association which can be used to retrieve the corresponding pattern with given (new) x or y. The only difference is that BAM adopts the polarized representation of binary data where $x_{ij}, y_{ij} \in \{-1, 1\}$ instead of binary representations using $\{1, 0\}$.

First proposed in [51], BAM use a simple matrix $M = \Sigma x_i^T y_i$ with given associations $\{x_i, y_i\}$ as input. Given x, the association pattern can be retrieved by simply using $\hat{y} = \mathcal{T}_\tau(Mx)$, where

$$\mathcal{T}_\tau(z) = \begin{cases} -1 & z <= -\tau \\ 0 & -\tau < z < \tau \\ 1 & z >= \tau \end{cases} \tag{6.2}$$

is an element-wise thresholding function. This type of BAM is called a non-iterative hetero-association memory. Whereas in optimal linear associative memory, $W = X^* Y$, defined as the multiplication of pseudo-inverse of input data X and output Y, is proved to perform better in terms of least square error [93]. In our empirical experiments, we didn't observe much difference in terms of retrieval ratio.

The main difference between hetero-association memory and pattern association problems comes from the input data. In BAM the input data are actual patterns instead of a large dataset D with hidden patterns $\mathcal{P}$. Even in the perfect scenario where $D = \mathcal{P}$, BAM still suffers from the well known problem of capacity: the internal matrix M has $d^{(L)} \times d^{(R)}$

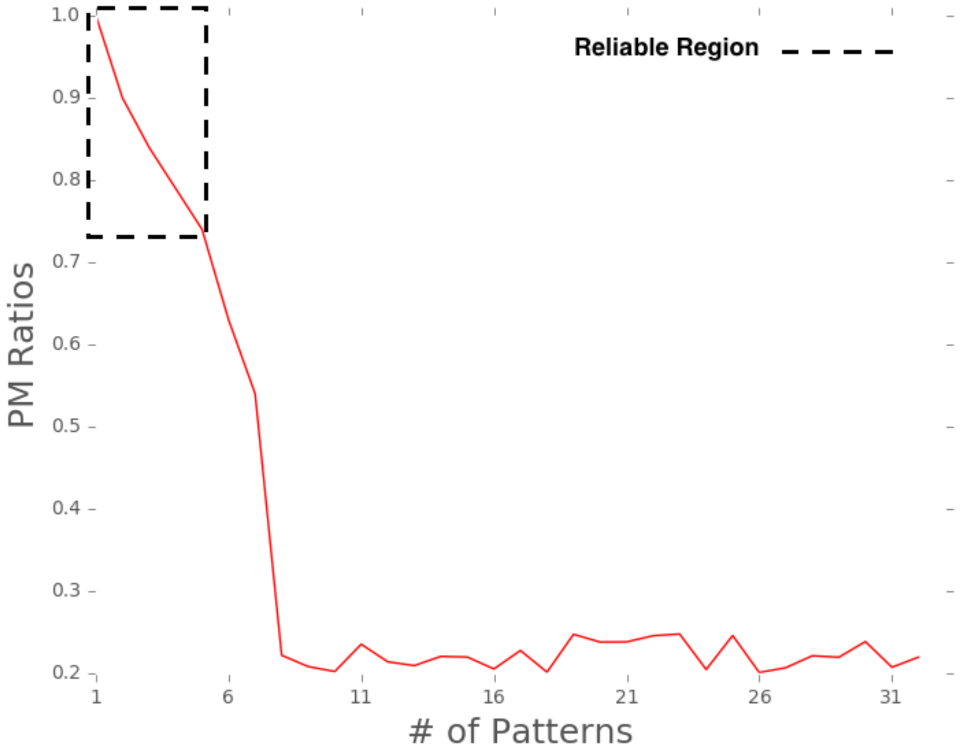


Fig. 6.1: Mean perfect matching ratio (10 trials) vs. number of patterns using BAM

degrees of freedom. Hence, BAM is able to reliably store and recall only $min(d^{(L)}, d^{(R)})$ independent vector pairs. As a sanity check, we performed experiments with simulated $D = \mathcal{P} = x_i, y_i$ of dimensions $d^{(L)} = 5, d^{(R)} = 5$. where x_i and y_i are randomly generated binary vectors satisfying Equation 6.2.3. As shown in Figure 6.1, the perfect matching ratio $\sum \mathbb{1}(\hat{y}_i = y_i)/|D|$ is decreasing with the increasing number of patterns $|D|$. In the annotated reliable region (the dashed rectangle) where $|D| \leq 5$, the performance is acceptable.

6.3 The Algorithm

In this section, we describe our proposed algorithm; its training and retrieval phases.

- **Training:** In the training phase our goal is to find possible label(s) for a given input vector x. In the dataset $D = \{x_i, y_i\}$ an input x may be associated with several sets of labels (vector y's) of which some occurrences are by chance and other correspond to ground truth. Our goal is to identify the true labels. A label y is considered significant (true) if its association with a given input x cannot be explained by random assignment; i.e., if the probability of observing a y with a given x by random assignment is very small.

Let $m = |D|$ denote the size of the entire training dataset and

$$p(x) = \frac{|\{x_i | x_i = x\}|}{m} \text{ and } p(y) = \frac{|\{y_i | y_i = y\}|}{m}$$

be the estimated probabilities (frequencies) of observing the vectors x and y respectively in the dataset, where x_i, y_i are binary vectors of dimensions $d^{(L)}$ and $d^{(R)}$ respectively. Similarly, let

$$p(x, y) = \frac{|\{(x_i, y_i) | x_i = x \wedge y_i = y\}|}{m}$$

be the joint probability of observing (x, y). Then, it can be seen that if the values of y are randomly associated with a given x, then the number of occurrences of a y should follow a binomial distribution[*] with probability of success $p(y)$ and associated number of trials $m \times p(x)$. We denote this random variable as a, and if the survival probability Prob$[a \geq p(x, y) \times m]$ is low, y_0 is significant. Weight matrix W^L assigns larger weights accordingly. In the reverse direction, using a similar concept, we define another weight matrix W^R which finds the desirable inputs associated with a given label vector.

We use the following two survival functions to quantify the confidence of any pair of patterns (x, y) by hypothesizing (H_0) that the observation of ratio $p(y|x)$ is a result of pure random selection of $p(x) \cdot m$ data points from $D^{(R)}$.

1.

$$W^{(L)}(x, y) = -\log\{\text{Prob}[a \geq p(x, y) \cdot m]\} \tag{6.3}$$

where a is a random variable with binomial distribution $B(a; p(x) \cdot m, p(y))$.

---

[*] The probability of observing the occurrences of $p(x, y)$ should follow a binomial distribution $B \sim (a; p(x)m, p(y))$.

2.

$$W^{(R)}(y,x) = -log\{\text{Prob}[b \geq p(x,y) \cdot m]\} \tag{6.4}$$

and b is drawn from the binomial distribution $B(b; p(y) \cdot m, p(x))$.

We use the survival function Prob$[a \geq p(x,y) \cdot m]$ to quantify the significance of a recognized pattern. As a result, if the probability $Prob[a \geq p(x,y) \cdot m]$ is small, then the null hypothesis is rejected with a high confidence, and we conclude that $x \rightarrow y$ is a promising pattern. The calculation of weight matrices in two directions is summarized in Algorithm 4.

Algorithm 4 Catworks Training

```
function TRAIN($D$)
    $m = |D|$                                          ▷ The size of dataset.
Step1:
    for all $x$ do
        Calculate the estimated probability $p(x)$
    for all $y$ do
        Calculate the estimated probability $p(y)$
    for all $(x, y)$ do
        Calculate the estimated joint probability $p(x, y)$
Step2:
    for all pair $(x, y)$ do
        (1) $W^{(L)}(x,y) = -\log[\text{Prob}\,(a \geq p(x,y) \times m)]$      ▷ see Equation 6.3

        (2) $W^{(R)}(x,y) = -\log[\text{Prob}\,(b \geq p(x,y) \times m)]$      ▷ see Equation 6.4

    return $W^{(L)}, W^{(R)}$
```

- **Retrieving – an iterative associative memory using top k targets:** In retrieval phase, we adopt an iterative method to keep updating two sets of patterns: $X^{(t)}, Y^{(t)}$ by selecting the "most relevant" k patterns using matrices created in the learning phase. Given with a query input x_0, we initialize the following quantities:

  - $X^{(0)} = \{x_0\}$, the initial singleton set.

  - $S^{(0)} = X^{(0)}$, the "core" of query patterns, which are incremented at each iteration.

  - Original score function $U(x_0) = 1$.

In each iteration, given with a set of x-values $X^{(i)}$, we assign scores to y using the following equation:

$$V(y) = \frac{\sum_{x \in X^{(i)}} W^{(L)}(x,y)U(x)}{|X^{(i)}|} \tag{6.5}$$

where high values of $V(y)$ implies more relevance between y and set $X^{(i)}$. We find a set $Y^{(i)}$ of size k such that

$$\forall_{y \in Y^{(i)},\, y' \notin Y^{(i)}}(V(y) \geq V(y')).$$

Reversely, we create the scores for input patterns using:

$$U(x) = \frac{\sum_{y \in Y^{(i)}} W^{(R)}(x,y)V(y)}{|Y^{(i)}|} \tag{6.6}$$

We find a set Z of size $k - |S^{(i)}|$ such that:

$$\forall_{x \in Z,\, x' \notin Z}(U(x) \geq U(x')).$$

Then we update $X^{(i+1)} = S^{(i)} \cup Z$ and $S^{(i+1)} = S^{(i)} \cup \{u\}$, where $u = \arg\max_x\{U(x)\}$. Essentially, $S(i) \subseteq X^{(i)}$ is considered as the core set of containing the original query x_0, which satisfies the following partial ordering property:

$$S^{(0)} \subseteq S^{(1)} \subseteq \ldots S^{(k-1)},$$

which grantees the maximum number iterations is less than or equal to k because the maximum size of core is k. In every iteration, we increment $S^{(i)}$ with the x highest score $U(x)$, this singleton is considered as the potential improved version of query x_0. As discussed in previous section, it is possible that the ground truth associative pattern $< x^*, y^* >$ may not have the highest scores[†]. So we adopt the iterative iteration to find a set of query input, which is considered as a set of variations of real pattern component x^*. As a result, by

---

[†] $\exists y' \neq y^* W^{(L)}(x^*, y') > W^{(L)}(x^*, y^*)$ or $\exists x' \neq x^* W^{(R)}(x', y^*) > W^{(R)}(x', y^*)$

Algorithm 5 Catworks Retrieval

```
function RETRIEVAL($x_0,\ W^{(L)},\ W^{(R)},\ k$)
    Initialize $S^{(0)} = X^{(0)} = \{x_0\}$, $U(x_0) = 1$, $i = 0$
    while $i \leq k$ and $Y^{(i+1)} \neq Y^{(i)}$ do
        1. Calculate the scores: $V(y)$                          ▷ see Equation 6.5
        2. Find the set $Y^{(i)}$ of size $k$ with top score in $V(y)$
        3. Calculate the scores: $U(x)$                          ▷ see Equation 6.6
        4. Find the set $Z$ of size $k - |S^{(i)}|$ with top score in $U(x)$
        5. $X^{(i+1)} = S^{(i)} \cup Z$
    return $Y^{(i)}$, in descending order of scores $V(y)$
```

using the final set $X^{(k)}$, we expect to find the real associative pattern component y^* using a weighted score from all x, where each x might have high scores for $y' \neq y^*$, but the overall score for y^* is maximized. The retrieval process is summarized in Algorithm 5.

6.4 Performance Evaluation

In this section, we evaluate the performance of the proposed algorithm for one synthetic dataset and few real datasets. We compare the performance of the proposed algorithm with existing algorithms. We use the perfect matching ratio

$$\frac{\sum \mathbb{1}(\hat{y}_i = y_i)}{|\mathcal{P}|}$$

and Hamming loss

$$\frac{\sum ||\hat{y}_i - y_i||}{|\mathcal{P}| \cdot d^{(R)}}$$

as our evaluation measures. Perfect matching is a very stringent evaluation criterion, where Hamming loss accepts "partially correct" answers.

6.4.1 Synthetic Data Generating

First of all, a set of associative patterns $\mathcal{P} = \{x_i : y_i\}$ is prepared. Recall that, by definition of a pattern, given input x_i we expect that the algorithm will output the vector y_i. Let $N_{\mathcal{P}}$ denote the distinct pairs in $\mathcal{P}$.

Let $\mathcal{P}' = \{x'_i : y'_i\}$ be the perturbed patterns. where

$$x'_{ij} = \neg x_{ij}, y'_{ij} = \neg y_{ij}$$

with a probability of β and

$$x'_{ij} = x_{ij}, y'_{ij} = y_{ij}$$

with a probability of $1 - \beta$.

Then we create a new dataset $D = \mathcal{P}' \cup D''$ where D'' is a set of random binary vectors of the same dimension. Thus, the set D represents a dataset consisting of true patterns along with some random terms.

6.4.2 Catworks performance analysis

In this first experiment, we study Catworks' performance. When $\alpha = 1$ and $\beta = 0$, it becomes the trivial case where a simple hashing function can be used to map the associative mapping. However, as discussed in previous section, the performance of BAM decreases when the number of stored patterns increases, due to the well-known capacity limit.

The final k vertices in $Y^{(t)}$ are the predicted associative patterns and V_y is the set of corresponding scores for them. In our evaluation, we use the best (highest score) as the final result. Results are shown in Figure 6.2 and Figure 6.3. It is worthwhile to note that Catworks is only sensitive to noise ratio β; thus, even if the ratio of planted pattern ratio

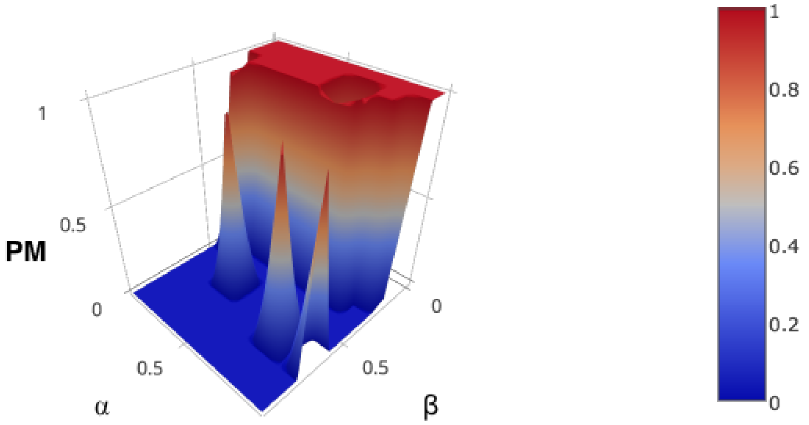


Fig. 6.2: Performance when $N = 1$.

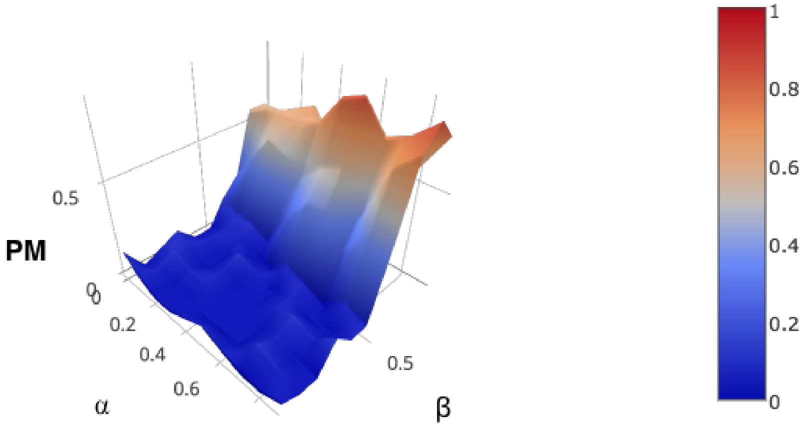


Fig. 6.3: Performance when $N = 11$.

α is small, Catworks can retrieve the associative patterns. In both experiments, we use the perfect matching ratio as the evaluation of performance.

We experimented Catworks with simulated data using multiple α, β, N values, for a comprehensive performance study.

6.4.3 Performance comparison

In this section, we discuss the results of performing a horizontal comparison with related approaches:

- Binary relevance based classifiers: (SVM[‡], logistic regression, Bayesian)
- BAM

In the following experiments, we fix $d^{(L)} = d^{(R)} = 15$ and $|D| = 20,000$ as the default parameters.[§] In the following experiments, we use different noise ratio values ($\beta = 0.15, 0.35$) to study the behavior of Catworks and other related approaches. In each

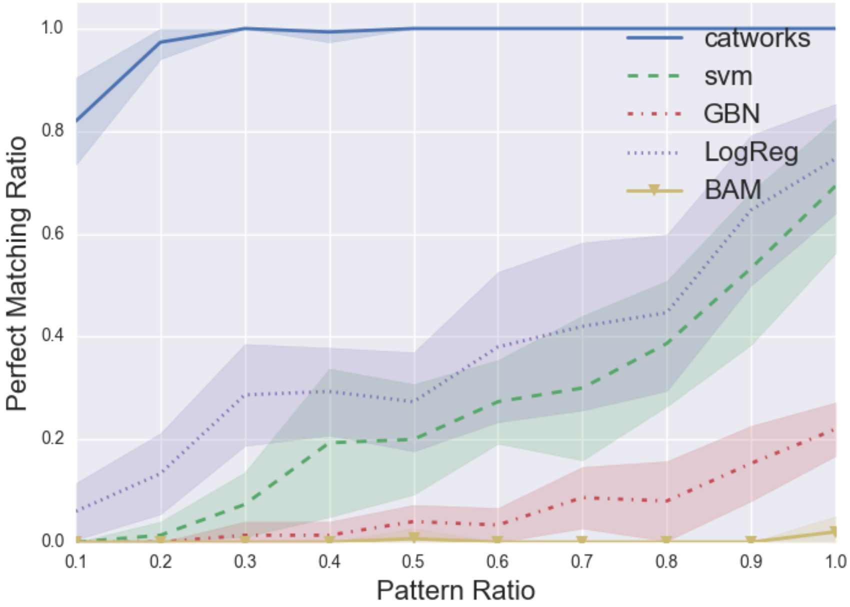


Fig. 6.4: Perfect matching vs. Pattern ratio ($\beta = 0.15$)

[‡] We used the bagging version of SVM with linear kernels.

[§] The potential number of distinct patterns is $2^{15} > |D|$. In our work, we also tested different parameter settings and observed similar results.

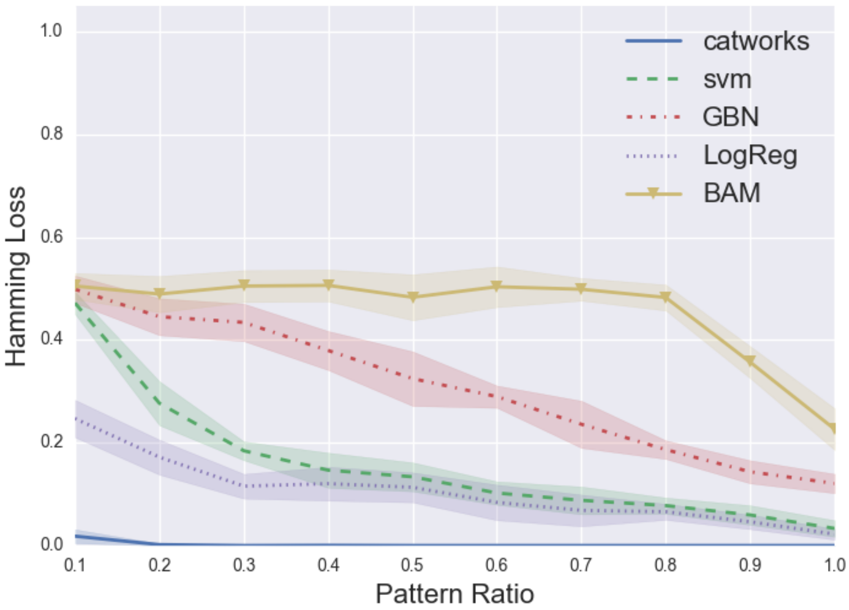


Fig. 6.5: Hamming loss v.s. Pattern ratio ($\beta = 0.15$)

experiment, we use pattern ratio $\alpha = [0.1, 1.0]$ to generate the synthetic datasets. For each configuration, 10 trials are repeated for a robust evaluation.

When $\beta = 0.15$ (Shown in Figure 6.5 and Figure 6.4), Catworks performs significantly better than other algorithms in both perfect matching ratio (higher is better) and Hamming loss (lower is better). In the plots, the shaded areas are $\pm\sigma$ in 10 trials of experiment, reflecting the robustness and stability of each algorithm. When $\beta = 0.35$, all algorithms perform worse compared with low noise ratio. However, Catworks no longer leads the performance. In Hamming loss evaluation, the retrieval performance of Catworks is only better than BAM (shown in Figure 6.7). In perfect matching evaluation (shown in Figure 6.6), Catworks still performs the best for low pattern ratios $\alpha \leq 0.5$. But for high pattern ratios, other algorithms (except BAM) performs better than Catworks.

This is a reasonable result, because the main advantage of Catworks is to learn the hidden patterns in $\mathcal{P}$, when noise ratio is high, the boundary or difference between $\mathcal{P}$ and $D - \mathcal{P}$ becomes marginal, then all the classification based methods perform better because they don not differentiate the real patterns from useless data. Classification based methods

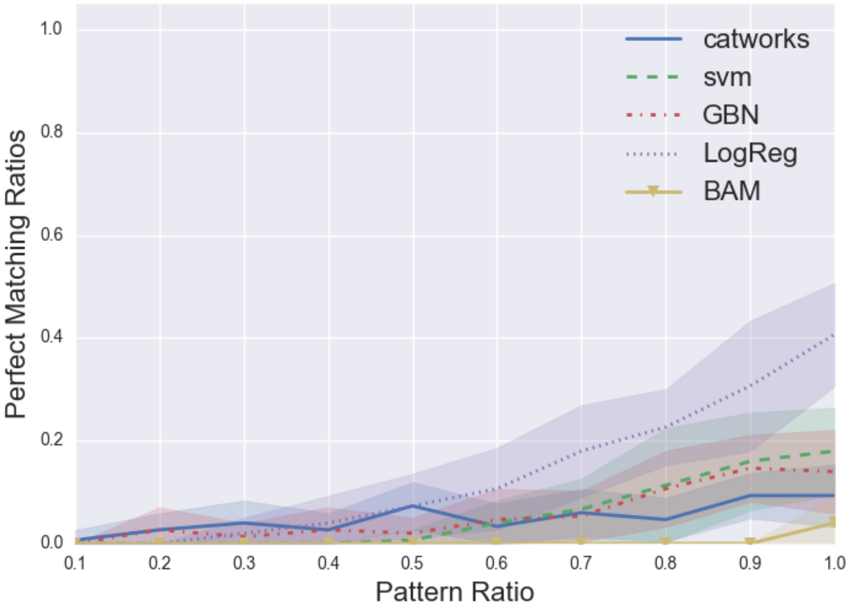


Fig. 6.6: Perfect matching vs. Pattern ratio ($\beta = 0.35$)

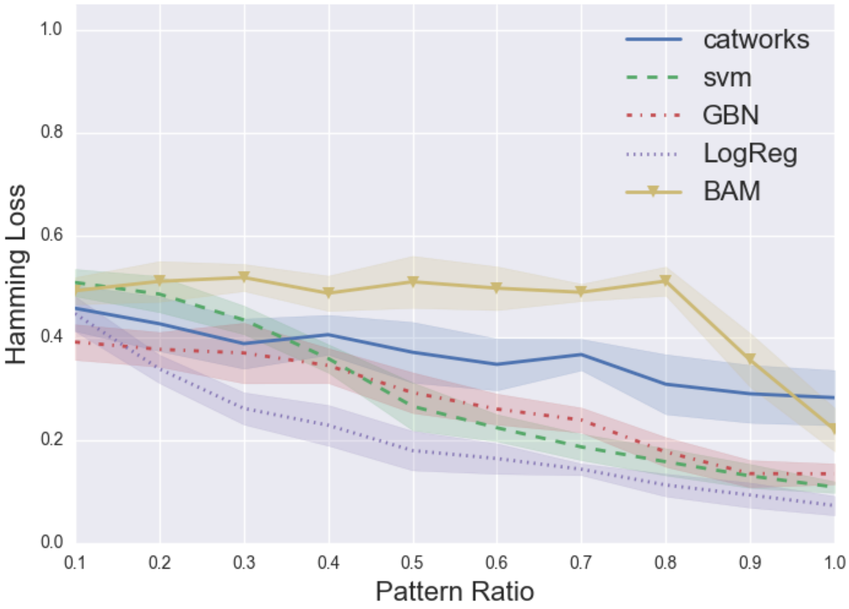


Fig. 6.7: Hamming loss vs. Pattern ratio ($\beta = 0.35$)

try to minimize the predicting error for the entire dataset D.

Another reason is Catworks' constraint (Equation 6.2.3), which requires at least one occurrence of $< x_i, y_i >$ in the dataset D. In high noise ratio scenarios, this is likely possible to be violated. As a result, Catworks will simply report nothing for x if x was never observed before. This is a true drawback of this algorithm. To address this problem, in such scenarios, we use the closest available neighbor in $M^{(L)}$ (in terms of Hamming distance) to x_0 as the delegate query input and call Algorithm 5 as usual.

6.4.4 Evaluation on real dataset

We also evaluated the performance of Catworks on real biological data from [96]. $\{x_i : y_i\}$ are histone modification and function annotations of genes where $x_{ij} = 1$ iff gene i has modification j active on it. where as $y_{ij} = 1$ iff gene i is labeled with function j. In the real data set; $d^{(L)} = 10$ is the number of different modifications and $d^{(R)} = 7$ is the number of annotations considered.

We selected the top 500 (in the descending order of support) as the ground truth patterns $\mathcal{P}$. Then we add random data following the same data generation procedure described in previous sections with parameter values $\alpha = 0.2, \beta = 0.15$. In 10 trials of experiments, Catworks achieved perfect matching ratio $\approx 0.83 \pm 0.12$ while all other algorithms achieved less than 0.5.

6.5 Concluding Remarks

In this chapter, we raised the problem of pattern association mining and compared with related similar problems. We also proposed a new algorithm, Catworks, which is designed to perform well even with high number of patterns. It performs better than other algorithms (multi-label classification based, content-addressable memory based) when noise ratio is reasonably low. However, when noise ratio is high ($\beta \geq 0.3$), Catworks performs worse than SVM and logistic regression.

This is because Catworks relies heavily on the quality of matrices $M^{(L)}$ and $M^{(R)}$ where real pattern pairs $\mathcal{P}_i =< x_i, y_i >$ are believed to have high values in both $M^{(L)}_{x_i,y_i}$ and $M^{(R)}_{y_i,x_i}$.

However, with high noise ratio, this assumption will collapse.

In future work, instead of creating such a one-to-one mapping matrices, we would like to improve Catworks with ideas from kernel based methods by creating a pairwise similarity network between different genes. In the retrieval phase, we may use the distance from a query input x to existing combinations to select multiple representatives instead of directly searching for exact matched patterns.

CHAPTER 7

CONCLUSION

First, we summarize the results and contributions of this study. Then we present some interesting future research directions for associative patterns recognition in biological regulations.

7.1 Summary

Associations describes the relations between objects, data and variables. In our brains, the concepts of objects are believed to associated with each other. In our daily lives, we can usually recall remotely related memories using relevant memories in terms of geological location, time or space. For example, a song played at someone's wedding triggers his/her detailed memories of the ceremony; a child experiences hunger at the sight of a logo of fast food restaurant. A central aspect of natural intelligence is that we seek to discover associations between different objects; this has also been used as the foundation of many artificial intelligence studies.

The classical associative patterns (or association patterns) recognition problem was first addressed mainly for supermarket data containing sets of items bought by customers, which are referred to as transactions. The original objective of this kind of analysis is to determine the associations between groups of items in transactions, which is essentially a many-to-many correlation between multiple items.

In our work, we generalized the idea of associative patterns to combinations between different universes of objects with heterogeneous data. Furthermore, we categorized the associations into "1:1", "n:1", "n:m" patterns. We proposed the problem of recognizing such patterns from multiple objects universes with heterogeneous data (categorical, textual, sequence or numeric data formats). To be specific, we focused on relating objects in biological regulation processes, i.e., genes, promoters, proteins, labels, histone modifiers.

First, we introduced an efficient algorithm to learn the protein-DNA binding relations between genes and promoters using an approach based on a modulo addition with hash function. As a result, we can efficiently recognize the target aptamer sequences with high affinity.

Then, we proposed a new transcription factor target scoring framework (TFTS) for gene regulatory network inference by incorporating target sequences survey in preliminary GRN. The refined output GRN is evaluated with available ground truth regulatory network, and we achieved up to 52% higher precision compared with ARACNe. This method provides a systematical approach for combining gene sequences and expression profiles for the first time also circumvented the challenge in indirect causal relations in GRN inference. The proposed method for "1:1" patterns (gene-gene patterns) also predicted new potential edges which are not available in the biological database, which can be used to help biologists to design new experiments.

Inspired by the "histone code" hypothesis, we proposed a new computational approach (HiPSiS) for "histone combination: biological function" associative patterns ("n:1" patterns) recognition. In this work, we added the market basket analysis to create a convolutional layer before applying classifiers directly on the histone modification data, and achieved 10% to 35% improvement in terms of precision. Additionally, HiPSiS also serve as a hypothesis generator to assign new labels to genes to address the scarcity of existing function annotations on genes. This approach was verified by the domain expert: in our proposed top 15 genes related to flowering-time, 14 of them were predicted to be candidates using an independent biological approach.

We extended HiPSiS to shape-based combinatorial patterns of histone modifications, and proposed an additional series compression and indexing step for efficient clustering. In an exploratory study, we studied the potential subclasses in flowering-time genes with different combinations of shapes, and observed that light-induced genes and non-light induced genes contain distinct combinations of patterns. This is novel compared to previous studies where only presence/absence of modification is considered.

Finally, we proposed an algorithm for discovering hidden "n:m" patterns from a noisy dataset. Our proposed information retrieval algorithm performs better than other approaches based on classification or content addressable associative memory.

7.2 Future Research Problems

The work in this dissertation can be extended by the following directions:

- **Increase Cardinality of Associative Patterns:** In this work, we limited the number of object universes to 2 for simplicity. In the future, it is worthwhile to study how to generalize the associative patterns across more universes ($L \geq 2$). For example, in histone modifications analysis, instead of relating only functional labels with histone modifications, we should investigate how to find the associative patterns among sequences, expression profiles and others.

This is crucial to understanding the fundamental mechanism of biological regulation given that the synergistic collaboration among multiple object universes (i.e., DNA, RNA, proteins, epigenetic regulators and external perturbation) ensures the fitness of complex systems such as living organisms. This is one of the most important objectives in computational biology, which seek to create the systematical "circuit logics" of the underlying mechanism in biological systems. KEGG [46] seeks to create such an encyclopedia of biological regulation by curating different sources of data (i.e., publications, existing models and manually uploaded data). However, it lacks the ability to infer associative patterns automatically, and requires extensive interactions with domain experts.

To extend the current work, future researchers can investigate the inter-layer link prediction problem in multilayer social networks for potential solutions, where layers represent different relations between users. Also, this is related to identity learning across different social networks, where IDs for the same user is not guaranteed to be identical, and the correct mapping of nodes from different layers is desired.

- **Dynamic Associative Patterns:** We have discussed the approaches to associative $n : m$ patterns, where the different combinations of objects (i.e., presence/absence of modifications, motifs and shapes) were considered as collaborating groups in a static snapshot of a biological systems. However, their temporal order are ignored in this work. For example, it is worthwhile to study the dynamic changes of such associative patterns across different stages of a biological organism (e.g., for plants, the biological regulation processes are believed to vary in different stages: budding, flowering, young and mature).

The real temporal histone modification data are limited, preventing such analysis at this time. However, more such dynamic epigenetic data will become available with advances in experimental equipment and processes.

- **Improve Catworks:** As mentioned in Chapter 6, Catworks cannot retrieve counterpart pattern y_i if x_i is not encountered due to noise or missing data, because we train the model to be a point-to-point associative pattern memory. Although we have addressed the problem using a delegating pattern available in $M^{(L)}$, the performance of Catworks relies heavily on the quality of input data. In the future work in this direction, one can investigate the potential of recognizing cluster-to-cluster associative patterns instead. Thus, whenever a query input x is not found in the relation matrix $M^{(L)}$, we can use its neighbors as delegates for prediction.

However, the construction of pattern neighborhoods is a non-trivial task because even if two combinations x, x^* are close in terms of Jaccard distance or cosine similarity (i.e., the composition of two nominal vectors are very similar), it is not guaranteed that corresponding patterns y, y^* are also close. In other words, Lebesgue continuity is not guaranteed:

$$|x - x^*| \leq \delta \not\Longrightarrow |y - y^*| \leq \epsilon.$$

To address this problem, future researchers should incorporate metric learning for nominal data. For example, the closeness of two vectors x, x^* should be determined by the comparing distribution of $p(y|x)$ and $p(y|x^*)$ using KL divergence.

- **Consider Additional Features:** As discussed in Chapter 1, biological regulation analysis is an area studying the complicated ensemble of various sub-systems. The understanding of how exactly living organisms work is limited at this time. To extend this work, one may explore the value of features such as: protein sequences, RNA expression, exon/intron annotations, SNPs and external perturbations in biological regulation.

Furthermore, feature engineering in bioinformatics research is a non-trivial task. In Chapter 5, we discussed the potential values of shape-based pattern recognition in gene function prediction. It is worthwhile to study different transformations of bioinformatics to improve the performance in associative pattern recognition.

Appendices

APPENDIX A

A.1 Proposed Sub-Classes of Flowering Genes

ID	Other Name	Description
AT1G01040	SUS1	SUSPENSOR 1
AT1G05830	SDG30	SET DOMAIN PROTEIN 30
AT1G08970	NF-YC9	"nuclear factor Y, subunit C9"
AT1G09570	PHYA	phytochrome A
AT1G10570	ULP1C	UB-LIKE PROTEASE 1C
AT1G14920	RGA2	RESTORATION ON GROWTH ON AMMONIA 2
AT1G26830	CUL3A	cullin 3A
AT1G28520	VOZ1	vascular plant one zinc finger protein
AT1G50700	CPK33	calcium-dependent protein kinase 33
AT1G51450	TRO	TRAUCO
AT1G53090	SPA4	SPA1-related 4
AT1G55250	HUB2	histone mono-ubiquitination 2
AT1G55325	MAB2	MACCHI-BOU 2
AT1G61040	VIP5	vernalization independence 5

AT1G62830	SWP1	
AT1G68050	FKF1	"flavin-binding, kelch repeat, f box 1"
AT1G71800	CSTF64	cleavage stimulating factor 64
AT1G72390	NA	
AT1G80070	SUS2	ABNORMAL SUSPENSOR 2
AT2G01570	RGA1	REPRESSOR OF GA1-3 1
AT2G02760	UBC2	ubiquiting-conjugating enzyme 2
AT2G17290	CPK6	calcium dependent protein kinase 6
AT2G18790	PHYB	phytochrome B
AT2G18915	LKP2	LOV KELCH protein 2
AT2G25930	PYK20	NA
AT2G30140	UGT87A2	UDP-glucosyl transferase 87A2
AT2G32950	FUS1	FUSCA 1
AT2G43010	SRL2	NA
AT2G44150	SDG7	SET DOMAIN-CONTAINING PROTEIN 7
AT2G44680	CKB4	casein kinase II beta subunit 4
AT2G46020	CHR2	CHROMATIN REMODELING 2
AT2G46260	LRB1	light-response BTB 1
AT2G46830	CCA1	circadian clock associated 1
AT3G03450	RGL2	RGA-like 2
AT3G10390	FLD	FLOWERING LOCUS D
AT3G12810	SRCAP	NA
AT3G20810	JMJD5	jumonji domain containing 5
AT3G22380	TIC	TIME FOR COFFEE

AT3G22590	PHP	PLANT HOMOLOGOUS TO PARAFIBROMIN
AT3G26640	LWD2	LIGHT-REGULATED WD 2
AT3G46640	PCL1	PHYTOCLOCK 1
AT3G48430	REF6	relative of early flowering 6
AT3G57300	INO80	INO80 ortholog
AT3G59060	PIL6	phytochrome interacting factor 3-like 6
AT3G63070	NA	NA
AT4G00650	RSB7	REDUCED STEM BRANCHING 7
AT4G00830	LIF2	LHP1-Interacting Factor 2
AT4G04920	SFR6	SENSITIVE TO FREEZING 6
AT4G10180	FUS2	FUSCA 2
AT4G15880	ESD4	EARLY IN SHORT DAYS 4
AT4G20400	PKDM7B	NA
AT4G24620	PGI1	phosphoglucose isomerase 1
AT4G29830	VIP3	vernalization independence 3
AT4G32980	ATH1	homeobox gene 1
AT4G34530	CIB1	cryptochrome-interacting basic-helix-loop-helix 1
AT4G37280	NA	NA
AT4G38680	GRP2	glycine rich protein 2
AT4G40060	HB16	homeobox protein 16
AT5G02810	PRR7	pseudo-response regulator 7
AT5G04240	ELF6	EARLY FLOWERING 6
AT5G13790	AGL15	AGAMOUS-like 15
AT5G15840	FG	NA

AT5G16320	FRL1	FRIGIDA like 1
AT5G24470	PRR5	pseudo-response regulator 5
AT5G47640	NF-YB2	"nuclear factor Y, subunit B2"
AT5G48300	APS1	ADP-GLUCOSE PYROPHOSPHORYLASE SMALL SUBUNIT 1
AT5G51230	VEF2	CYTOKININ RESISTANT 1
AT5G58230	MSI1	MULTICOPY SUPRESSOR OF IRA1
AT5G60120	TOE2	target of early activation tagged (EAT) 2
AT5G63110	SIL1	NA
AT5G65060	MAF3	MADS AFFECTING FLOWERING 3

Table A.1: The detailed list of genes with combination of shapes "LLL".

ID	Other Name	Description
AT1G02400	GA2OX6	gibberellin 2-oxidase 6
AT1G04400	PHH1	NA
AT1G15550	GA4	GA REQUIRING 4
AT1G25540	PFT1	PHYTOCHROME AND FLOWERING TIME 1
AT1G47990	GA2OX4	gibberellin 2-oxidase 4
AT1G50960	GA2OX7	gibberellin 2-oxidase 7
AT1G51140	FBH3	FLOWERING BHLH 3
AT1G57820	VIM1	VARIANT IN METHYLATION 1
AT1G66050	VIM2	VARIANT IN METHYLATION 2
AT1G66650	NA	
AT1G69120	AP1	APETALA1
AT1G76710	SDG26	SET domain group 26

AT1G79430	WDY	WOODY
AT1G79730	ELF7	EARLY FLOWERING 7
AT2G03500	NA	NA
AT2G19425	MIR156G	microRNA156G
AT2G21660	GRP7	GLYCINE-RICH RNA-BINDING PROTEIN 7
AT2G22630	AGL17	AGAMOUS-like 17
AT2G25095	MIR156A	microRNA156A
AT2G27550	ATC	centroradialis
AT2G28550	TOE1	TARGET OF EARLY ACTIVATION TAGGED (EAT) 1
AT2G31650	SDG27	SET DOMAIN PROTEIN 27
AT2G33835	FES1	FRIGIDA-ESSENTIAL 1
AT2G34880	PKDM7C	NA
AT2G44950	RDO4	REDUCED DORMANCY 4
AT2G45650	RSB1	REDUCED SHOOT BRANCHING 1
AT2G45660	SOC1	SUPPRESSOR OF OVEREXPRESSION OF CO 1
AT2G46340	SPA1	SUPPRESSOR OF PHYA-105 1
AT3G01090	SNRK1.1	SNF1-RELATED PROTEIN KINASE 1.1
AT3G04610	FLK	flowering locus KH domain
AT3G11910	UBP13	ubiquitin-specific protease 13
AT3G20740	FIS3	NA
AT3G23060	NA	NA
AT3G24440	VRN5	VERNALIZATION 5

AT3G43190	SUS4	sucrose synthase 4
AT3G44110	J3	NA
AT3G45880	NA	NA
AT3G46510	PUB13	plant U-box 13
AT3G49600	UBP26	ubiquitin-specific protease 26
AT3G54500	LNK2	night light-inducible and clock-regulated 2
AT3G54560	HTA11	histone H2A 11
AT3G54990	SMZ	SCHLAFMUTZE
AT4G10710	SPT16	global transcription factor C
AT4G11880	AGL14	AGAMOUS-like 14
AT4G16250	PHYD	phytochrome D
AT4G16280	FCA	NA
AT4G20370	TSF	TWIN SISTER OF FT
AT4G21200	GA2OX8	gibberellin 2-oxidase 8
AT4G22140	EBS	EARLY BOLTING IN SHORT DAYS
AT4G23100	RML1	ROOT MERISTEMLESS 1
AT4G25420	GA5	GA REQUIRING 5
AT4G26440	WRKY34	WRKY DNA-binding protein 34
AT4G30200	VIL2	VIN3-Like 2
AT5G03840	TFL1	TERMINAL FLOWER 1
AT5G04275	MIR172B	microRNA172B
AT5G07200	YAP169	NA
AT5G09740	HAM2	histone acetyltransferase of the MYST family 2
AT5G10140	RSB6	REDUCED STEM BRANCHING 6

AT5G10945	MIR156D	microRNA156D
AT5G11977	MIR156E	microRNA156E
AT5G13480	FY	
AT5G17690	TFL2	TERMINAL FLOWER 2
AT5G24860	FPF1	FLOWERING PROMOTING FACTOR 1
AT5G26147	MIR156F	microRNA156F
AT5G35910	NA	NA
AT5G39550	VIM3	VARIANT IN METHYLATION 3
AT5G42400	SDG25	SET domain protein 25
AT5G44200	CBP20	CAP-binding protein 20
AT5G46210	CUL4	cullin4
AT5G48890	LATE	LATE FLOWERING
AT5G51810	GA20OX2	gibberellin 20 oxidase 2
AT5G51820	STF1	STARCH-FREE 1
AT5G61060	HDA5	NA
AT5G61150	VIP4	VERNALIZATION INDEPENDENCE 4
AT5G62040	BFT	brother of FT and TFL1
AT5G65070	MAF4	MADS AFFECTING FLOWERING 4

Table A.2: The detailed list of genes with combination of shapes "VVV".

A.2 Siginficant Combinations of Histone Modifications for Functions

Function*	Top patterns	Bottom patterns
Stress	H3K18AC+H3K9AC H3K4ME3 H3K18AC+H3K9AC+H3K36ME3 H3K18AC+H3K4ME3+H3K9AC+H3K36ME3 H3K4ME3+H3K9AC	H3K27ME1 H3 H3K27ME1+H3 H3K9ME2 H3K9ME2+H3+H3K27ME1
Stimulus	H3K9AC H3K4ME3+H3K9AC H3K4ME3 H3K9AC+H3K36ME3 H3K4ME3+H3K9AC+H3K36ME3	H3 H3K27ME1 H3K9ME2 H3K27ME1+H3 H3K27ME3
Development	H3K36ME2 H3K27ME3 H3K27ME3+H3K9AC+H3K36ME3 H3K27ME3+H3K4ME3+H3K36ME3 H3K27ME3+H3K4ME3+H3K4ME2+H3K36ME3	H3 H3K27ME1 H3K18AC H3K27ME1+H3 H3K9ME2
Defense	H3K9AC H3K18AC+H3K9AC H3K4ME3+H3K9AC H3K18AC H3K18AC+H3K4ME3	H3K27ME1 H3K9ME2 H3 H3K36ME3+H3 H3K4ME3+H3K36ME3+H3
Flowering	H3K4ME3 H3K4ME3+H3K36ME3 H3K36ME3 H3K4ME3+H3K4ME2+H3K36ME3 H3K4ME3+H3K9AC	H3K27ME3 H3K27ME3+H3K36ME3 H3K18AC H3 H3K27ME3+H3K4ME3

Table A.3: Function-specific combinatorial histone modification patterns. Functions labeling (marked with asterisk) are defined and obtained from TAIR gene ontology database [7]. Top patterns are desired modifications in the specific function, whereas bottom combinations are considered as inhibitors.

A.3 Extended Study of HiPSiS with Other Labels

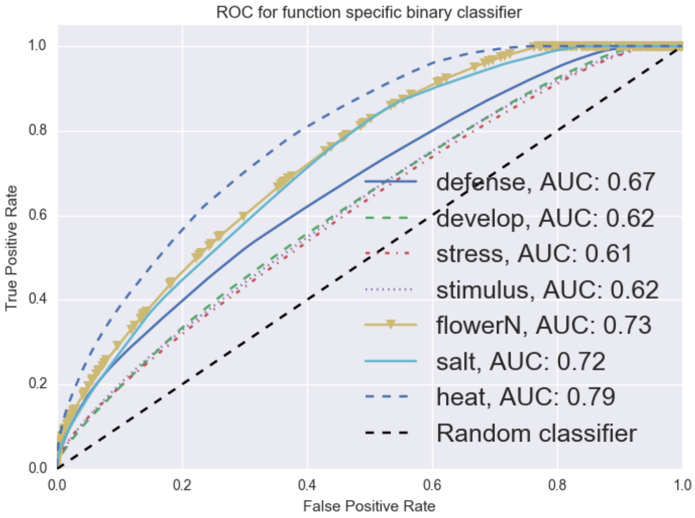


Fig. A.1: ROC curves of HiPSiS with additional heat and salt stress.

A.4 Details of Haar Transformation

Decomposition: The low (H_0) and high (H_1) pass filters (Haar wavelet) for data series of length $n = 2^k$ are defined as follows:

$$H_0^{(n)} = \frac{1}{2}\begin{bmatrix} 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & & \ddots & & \\ & & & & 1 & 1 \\ & & & & & 1 \end{bmatrix} \in R^{n\times n}$$

and

$$H_1^{(n)} = \frac{1}{2}\begin{bmatrix} 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & & \ddots & & \\ & & & & 1 & -1 \\ & & & & & 1 \end{bmatrix} \in R^{n\times n}$$

In Figure 5.4, the circled down-arrow represent the down-sampling of input. Let

$$D^{(n)} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix} \in R^{\frac{n}{2}\times n}$$

denote the down sampling matrix. For convenience, we use

$$L^{(n)} = D^{(n)} H_0^{(n)}$$

and

$$B^{(n)} = D^{(n)} H_1^{(n)}$$

to denote the calculation of approximation and detail coefficients in each iteration. As a results, we obtain coefficients of different lengths (i.e., $1, 1, 2, 4, \dots$):

$$w = L^{(1)} L^{(2)} \dots L^{(n)} x,\ B^{(1)} L^{(2)} \dots L^{(n)} x,\ \dots, B^{(n)} x,$$

and the complete Haar transformation for length n is defined as

$$H^{(n)} = \begin{bmatrix} L^{(n)} \\ — \\ B^{(n)} \end{bmatrix} \in R^{n \times n}.$$

Reconstruction: In this process, we use the output of the previous decomposition step to reconstruct a the approximation vector $\tilde{x}$. This process can be viewed as the reverse procedure of decomposition. In Haar wavelet example, the first iteration will use coefficients $w_1 = L^{(1)} L^{(2)} \dots L^{(n)} x$ and $w_2 = B^{(1)} L^{(2)} \dots L^{(n)} x$ as the inputs to the reconstruction part.

$$\tilde{x}^{(2)} = (H^{(2)})^T \begin{bmatrix} w_1 \\ — \\ w_2 \end{bmatrix}$$

Iteratively, $\tilde{x}^{(4)}$ can be calculated using:

$$\tilde{x}^{(4)} = (H^{(4)})^T \begin{bmatrix} \tilde{x}^{(2)} \\ — \\ w_3 \end{bmatrix}.$$

REFERENCES

[1] D. Abdulrehman, P. T. Monteiro, M. C. Teixeira, N. P. Mira, A. B. Lourenço, S. C. dos Santos, T. R. Cabrito, A. P. Francisco, S. C. Madeira, R. S. Aires, A. L. Oliveira, I. Sá-Correia, and A. T. Freitas, "YEASTRACT: providing a programmatic access to curated transcriptional regulatory associations in Saccharomyces cerevisiae through a web services interface." *Nucleic acids research*, vol. 39, no. Database issue, pp. D136–40, Jan. 2011.

[2] M. Adedoyin-Olowe, M. M. Gaber, and F. Stahl, "A survey of data mining techniques for social media analysis," *arXiv preprint arXiv:1312.4617*, 2013.

[3] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *Journal of molecular biology*, vol. 215, no. 3, pp. 403–410, 1990.

[4] T. Bahadori, "Non-parametric entropy estimation," 2012.

[5] M. A. Beer and S. Tavazoie, "Predicting gene expression from sequence," *Cell*, vol. 117, no. 2, pp. 185–198, 2004.

[6] S. Bennett, "Solexa ltd," *Pharmacogenomics*, vol. 5, no. 4, pp. 433–438, 2004.

[7] T. Z. Berardini, L. Reiser, D. Li, Y. Mezheritsky, R. Muller, E. Strait, and E. Huala, "The arabidopsis information resource: Making and mining the gold standard annotated reference plant genome," *genesis*, vol. 53, no. 8, pp. 474–485, 2015.

[8] C. Bernard, *Introduction à l'étude de la médecine expérimentale par m. Claude Bernard*. Baillière, 1865.

[9] A. Berr, R. Ménard, T. Heitz, and W.-H. Shen, "Chromatin modification and remodelling: a regulatory landscape for the control of arabidopsis defence responses upon pathogen attack," *Cellular microbiology*, vol. 14, no. 6, pp. 829–839, 2012.

[10] A. Berr, S. Shafiq, and W.-H. Shen, "Histone modifications in transcriptional activation during plant development," *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms*, vol. 1809, no. 10, pp. 567–576, 2011.

[11] L. Bich, M. Mossio, K. Ruiz-Mirazo, and A. Moreno, "Biological regulation: controlling the system from within," *Biology & Philosophy*, vol. 31, no. 2, pp. 237–265, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10539-015-9497-8

[12] C. M. Bishop, "Pattern recognition," *Machine Learning*, vol. 128, pp. 210–212, 2006.

[13] G. E. Box, W. G. Hunter, and J. S. Hunter, *Statistics for experimenters: an introduction to design, data analysis, and model building*. JSTOR, 1978, vol. 1.

[14] J. A. Brusslan, G. Bonora, A. M. Rus-Canterbury, F. Tariq, A. Jaroszewicz, and M. Pellegrini, "A genome-wide chronological study of gene expression and two histone modifications, h3k4me3 and h3k9ac, during developmental leaf senescence," *Plant physiology*, vol. 168, no. 4, pp. 1246–1261, 2015.

[15] L. Bülow, J. C. Bolívar, J. Ruhe, Y. Brill, and R. Hehl, "'MicroRNA Targets', a new AthaMap web-tool for genome-wide identification of miRNA targets in Arabidopsis thaliana." *BioData mining*, vol. 5, no. 1, p. 7, 2012.

[16] M. L. Bulyk, A. M. McGuire, N. Masuda, and G. M. Church, "A motif co-occurrence approach for genome-wide prediction of transcription-factor-binding sites in Escherichia coli." *Genome research*, vol. 14, no. 2, pp. 201–208, Feb. 2004.

[17] C. J. Burges, “A tutorial on support vector machines for pattern recognition,” *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[18] J.-B. F. Charron, H. He, A. A. Elling, and X. W. Deng, “Dynamic landscapes of four histone modifications during deetiolation in arabidopsis,” *The Plant Cell*, vol. 21, no. 12, pp. 3732–3748, 2009.

[19] C. P. Chen and C.-Y. Zhang, “Data-intensive applications, challenges, techniques and technologies: A survey on big data,” *Information Sciences*, vol. 275, pp. 314–347, 2014.

[20] P. Choudhary, P. Saha, T. Ray, Y. Tang, D. Yang, and M. C. Cannon, “Extensin18 is required for full male fertility as well as normal vegetative growth in arabidopsis,” *Frontiers in Plant Science*, vol. 6, p. 553, 2015. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4510346/

[21] C. E. Cook, M. T. Bergman, R. D. Finn, G. Cochrane, E. Birney, and R. Apweiler, “The european bioinformatics institute in 2016: Data growth and integration,” *Nucleic Acids Research*, vol. 44, no. D1, p. D20, 2016. [Online]. Available: +http://dx.doi.org/10.1093/nar/gkv1352

[22] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[23] K. Daily, V. R. Patel, P. Rigor, X. Xie, and P. Baldi, “MotifMap: integrative genome-wide maps of regulatory motif sites for model species.” *BMC bioinformatics*, vol. 12, p. 495, 2011.

[24] R. V. Davuluri, H. Sun, S. K. Palaniswamy, N. Matthews, C. Molina, M. Kurtz, and E. Grotewold, “Agris: Arabidopsis gene regulatory information server, an information resource of arabidopsis cis-regulatory elements and transcription factors,” *BMC Bioinformatics 2003, 4:25*, 2003.

[25] G. G. Della Gatta, M. M. Bansal, A. A. Ambesi-Impiombato, D. D. Antonini, C. C. Missero, and D. D. di Bernardo, "Direct targets of the TRP63 transcription factor revealed by a combination of gene expression profiling and reverse engineering." *Genes & Development*, vol. 18, no. 6, pp. 939–948, Jun. 2008.

[26] B. Ding and G.-L. Wang, "Chromatin versus pathogens: the function of epigenetics in plant immunity," *Frontiers in plant science*, vol. 6, 2015.

[27] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences*, vol. 95, no. 25, pp. 14 863–14 868, 1998.

[28] J. Ernst and M. Kellis, "Discovery and characterization of chromatin states for systematic annotation of the human genome," *Nat Biotech*, vol. 28, no. 8, pp. 817–825, 08 2010. [Online]. Available: http://dx.doi.org/10.1038/nbt.1662

[29] M. Famulok, J. S. Hartig, and G. Mayer, "Functional aptamers and aptazymes in biotechnology, diagnostics, and therapy." *Chemical reviews*, vol. 107, no. 9, pp. 3715–3743, Sep. 2007.

[30] C. Flensburg, S. A. Kinkel, A. Keniry, M. E. Blewitt, and A. Oshlack, "A comparison of control samples for chip-seq of histone modifications," *Frontiers in Genetics*, vol. 5, p. 329, 2014. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4174756/

[31] A. J. Gnanam, B. Hall, X. Shen, S. Piasecki, A. Vernados, E. E. Galyov, S. J. Smither, G. B. Kitto, R. W. Titball, A. D. Ellington, and K. A. Brown, "Development of aptamers specific for potential diagnostic targets in Burkholderia pseudomallei." *Transactions of the Royal Society of Tropical Medicine and Hygiene*, vol. 102 Suppl 1, pp. S55–S57, Dec. 2008.

[32] R. W. Hamming, "Error detecting and error correcting codes," *Bell Labs Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.

[33] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *ACM Sigmod Record*, vol. 29, no. 2. ACM, 2000, pp. 1–12.

[34] H. D. Herce, W. Deng, J. Helma, H. Leonhardt, and M. C. Cardoso, "Visualization and targeted disruption of protein interactions in living cells," *Nature communications*, vol. 4, 2013.

[35] C. Hidber, *Online association rule mining*. ACM, 1999, vol. 28, no. 2.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[37] M. M. Hoffman, J. Ernst, S. P. Wilder, A. Kundaje, R. S. Harris, M. Libbrecht, B. Giardine, P. M. Ellenbogen, J. A. Bilmes, E. Birney *et al.*, "Integrative annotation of chromatin elements from encode data," *Nucleic acids research*, p. gks1284, 2012.

[38] F. Iglesias and W. Kastner, "Analysis of similarity measures in times series clustering for the discovery of building energy patterns," *Energies*, vol. 6, no. 2, pp. 579–597, 2013.

[39] F. Jacob, *La logique du vivant: une histoire de l'hérédité*. Gallimard, 1987.

[40] F. Jacob and J. Monod, "Genetic regulatory mechanisms in the synthesis of proteins," *Journal of molecular biology*, vol. 3, no. 3, pp. 318–356, 1961.

[41] T. Jenuwein and C. D. Allis, "Translating the histone code," *Science*, vol. 293, no. 5532, pp. 1074–1080, 2001.

[42] H. J. Jeong, J. Yang, J. Yi, and G. An, "Controlling flowering time by histone methylation and acetylation in arabidopsis and rice," *Journal of Plant Biology*, vol. 58, no. 4, pp. 203–210, 2015.

[43] D. S. Johnson, A. Mortazavi, R. M. Myers, and B. Wold, “Genome-wide mapping of in vivo protein-dna interactions,” *Science*, vol. 316, no. 5830, pp. 1497–1502, 2007. [Online]. Available: http://science.sciencemag.org/content/316/5830/1497

[44] R. J. S. Jr., M. T. Koobatian, A. Shahini, D. D. Swartz, and S. T. Andreadis, “Capture of endothelial cells under flow using immobilized vascular endothelial growth factor,” *Biomaterials*, vol. 51, pp. 303 – 312, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0142961215001350

[45] B. K, A. Margolin, G. Stolovitzky, U. Klein, R. Dalla-Favera, and A. Califano, “Reverse engineering of regulatory networks in human b cells,” *Nat Genet*, vol. 37(4), pp. 382–390, 2005.

[46] M. Kanehisa and S. Goto, “Kegg: kyoto encyclopedia of genes and genomes,” *Nucleic acids research*, vol. 28, no. 1, pp. 27–30, 2000.

[47] N. Khan, I. Yaqoob, I. A. T. Hashem, Z. Inayat, W. K. Mahmoud Ali, M. Alam, M. Shiraz, and A. Gani, “Big data: survey, technologies, opportunities, and challenges,” *The Scientific World Journal*, vol. 2014, 2014.

[48] J.-M. Kim, T. Sasaki, M. Ueda, K. Sako, and M. Seki, “Chromatin changes in response to drought, salinity, heat, and cold stresses in plants,” *Frontiers in plant science*, vol. 6, p. 114, 2015.

[49] R. P. Kindermann and J. L. Snell, “On the relation between markov random fields and social networks,” *Journal of Mathematical Sociology*, vol. 7, no. 1, pp. 1–13, 1980.

[50] Y. Kondo, L. Shen, P. S. Yan, T. H.-M. Huang, and J.-P. J. Issa, “Chromatin immunoprecipitation microarrays for identification of genes silenced by histone h3 lysine 9 methylation,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 19, pp. 7398–7403, 2004.

[51] B. Kosko, “Bidirectional associative memories,” *IEEE Transactions on Systems, man, and Cybernetics*, vol. 18, no. 1, pp. 49–60, 1988.

[52] T. Kouzarides, “Chromatin modifications and their function,” *Cell*, vol. 128, no. 4, pp. 693–705, 2007.

[53] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[54] G. V. Kupakuwana, J. E. Crill, II, M. P. McPike, and P. N. Borer, “Acyclic identification of aptamers for human alpha-thrombin using over-represented libraries and deep sequencing,” *PLoS ONE*, vol. 6, no. 5, p. e19395, 05 2011.

[55] C. Lainscsek and T. J. Sejnowski, “Delay differential analysis of time series,” *Neural computation*, 2015.

[56] P. Lamesch, T. Z. Berardini, D. Li, D. Swarbreck, C. Wilks, R. Sasidharan, R. Muller, K. Dreher, D. L. Alexander, M. Garcia-Hernandez, A. S. Karthikeyan, C. H. Lee, W. D. Nelson, L. Ploetz, S. Singh, A. Wensel, and E. Huala, “The Arabidopsis Information Resource (TAIR): improved gene annotation and new tools.” *Nucleic acids research*, vol. 40, no. Database issue, pp. D1202–10, Jan. 2012.

[57] T. W. Liao, “Clustering of time series dataâĂŤa survey,” *Pattern recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[58] C. Linghu, H. Zheng, L. Zhang, and J. Zhang, “Discovering common combinatorial histone modification patterns in the human genome,” *Gene*, vol. 518, no. 1, pp. 171–178, 2013.

[59] C. Luo, D. J. Sidote, Y. Zhang, R. A. Kerstetter, T. P. Michael, and E. Lam, “Integrative analysis of chromatin states in arabidopsis identified potential regulatory

mechanisms for natural antisense transcript production," *The Plant Journal*, vol. 73, no. 1, pp. 77–90, 2013.

[60] N. M. Luscombe, D. Greenbaum, M. Gerstein *et al.*, "What is bioinformatics? an introduction and overview," *Yearbook of Medical Informatics*, vol. 1, no. 83-100, p. 2, 2001.

[61] S. Ma, Q. Gong, and H. J. Bohnert, "An Arabidopsis **gene net**work based on the graphical Gaussian model." *Genome research*, vol. 17, no. 11, pp. 1614–1625, Nov. 2007.

[62] L. T. MacNeil and A. J. Walhout, "Gene regulatory networks and the role of robustness and stochasticity in the control of gene expression," *Genome research*, vol. 21, no. 5, pp. 645–657, 2011.

[63] A. A. A. Margolin, I. I. Nemenman, K. K. Basso, C. C. Wiggins, G. G. Stolovitzky, R. R. D. Favera, and A. A. Califano, "ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context." *BMC Bioinformatics*, vol. 7 Suppl 1, pp. S7–S7, Jan. 2006.

[64] A. Medina-Rivera, O. Sand, and C. Herrmann, "RSAT 2011: regulatory sequence analysis tools," … *acids research*, 2011.

[65] M. Meyerson, S. Gabriel, and G. Getz, "Advances in understanding cancer genomes through second-generation sequencing." *Nature reviews. Genetics*, vol. 11, no. 10, pp. 685–696, Oct. 2010.

[66] J. Monod, J.-P. Changeux, and F. Jacob, "Allosteric proteins and cellular control systems," *Journal of molecular biology*, vol. 6, no. 4, pp. 306–329, 1963.

[67] A. Mortazavi, S. Pepke, C. Jansen, G. K. Marinov, J. Ernst, M. Kellis, R. C. Hardison, R. M. Myers, and B. J. Wold, "Integrating and mining the chromatin landscape of

cell-type specificity using self-organizing maps," *Genome research*, vol. 23, no. 12, pp. 2136–2148, 2013.

[68] K. P. Murphy, "Dynamic bayesian networks," *Probabilistic Graphical Models, M. Jordan*, vol. 7, 2002.

[69] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.

[70] U. Ozertem and D. Erdogmus, "Principal curve time warping," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2041–2049, 2009.

[71] T. Paixão and R. B. R. Azevedo, "Redundancy and the evolution of cis-regulatory element multiplicity." *PLoS computational biology*, vol. 6, no. 7, p. e1000848, 2010.

[72] A. R. Quinlan and I. M. Hall, "Bedtools: a flexible suite of utilities for comparing genomic features," *Bioinformatics*, vol. 26, no. 6, pp. 841–842, 2010.

[73] O. J. Rando, "Combinatorial complexity in chromatin structure and function: revisiting the histone code," *Current opinion in genetics & development*, vol. 22, no. 2, pp. 148–155, 2012.

[74] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine learning*, vol. 85, no. 3, pp. 333–359, 2011.

[75] S. Reuter, S. C. Gupta, B. Park, A. Goel, and B. B. Aggarwal, "Epigenetic changes induced by curcumin and other natural compounds," *Genes & nutrition*, vol. 6, no. 2, pp. 93–108, 2011.

[76] J.-J. M. Riethoven, "Regulatory regions in dna: promoters, enhancers, silencers, and insulators," *Computational Biology of Transcription Factor Binding*, pp. 33–42, 2010.

[77] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complementary dna microarray,"

Science, vol. 270, no. 5235, pp. 467–470, 1995. [Online]. Available: http://science.sciencemag.org/content/270/5235/467

[78] D. E. Schones and K. Zhao, “Genome-wide approaches to studying chromatin modifications,” *Nature Reviews Genetics*, vol. 9, no. 3, pp. 179–191, 2008.

[79] D. Schübeler, D. M. MacAlpine, D. Scalzo, C. Wirbelauer, C. Kooperberg, F. Van Leeuwen, D. E. Gottschling, L. P. O’Neill, B. M. Turner, J. Delrow *et al.*, “The histone modification pattern of active genes revealed through genome-wide chromatin analysis of a higher eukaryote,” *Genes & development*, vol. 18, no. 11, pp. 1263–1271, 2004.

[80] H. Sheng, K. Mehrotra, C. Mohan, and R. Raina, “Hammer algorithm: Hashing with arithmetic modulo-4 for motif extraction of regulatory elements,” in *Bioinformatics and Bioengineering, 2007. BIBE 2007. Proceedings of the 7th IEEE International Conference on*, 2007, pp. 753–758.

[81] H. Sheng, K. Mehrotra, C. K. Mohan, and R. Raina, “HAMMER algorithm: Hashing with arithmetic modulo-4 for motif extraction of regulatory elements,” in *BIBE*. IEEE, 2007, pp. 753–758. [Online]. Available: http://dx.doi.org/10.1109/BIBE.2007.4375645

[82] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, “The hadoop distributed file system,” in *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, May 2010, pp. 1–10.

[83] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher, “Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization.” *Molecular biology of the cell*, vol. 9, no. 12, pp. 3273–3297, Dec. 1998.

[84] R. Stoltenburg, C. Reinemann, and B. Strehlitz, “SELEX—a (r) evolutionary method to generate high-affinity nucleic acid ligands,” *Biomolecular engineering*, 2007.

[85] B. D. Strahl and C. D. Allis, “The language of covalent histone modifications,” *Nature*, vol. 403, no. 6765, pp. 41–45, 01 2000. [Online]. Available: http://dx.doi.org/10.1038/47412

[86] B. Taskar, V. Chatalbashev, and D. Koller, “Learning associative markov networks,” in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 102.

[87] G. Tsoumakas and I. Katakis, “Multi-label classification: An overview,” *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, 2006.

[88] UCSC. (2010) Ucsc description of track files. [Online]. Available: https://genome.ucsc.edu/goldenPath/help/wiggle.html

[89] S. A. Vlahopoulos, O. Cen, N. Hengen, J. Agan, M. Moschovi, E. Critselis, M. Adamaki, F. Bacopoulou, J. A. Copland, I. Boldogh, M. Karin, and G. P. Chrousos, “Dynamic aberrant nf-Îžb spurs tumorigenesis: A new model encompassing the microenvironment,” *Cytokine & Growth Factor Reviews*, vol. 26, no. 4, pp. 389 – 403, 2015, sI:Cytokines and growth factors in cancer biology. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1359610115000477

[90] C. R. Vogel and M. E. Oman, “Iterative methods for total variation denoising,” *SIAM Journal on Scientific Computing*, vol. 17, no. 1, pp. 227–238, 1996.

[91] R. A. Wagner and M. J. Fischer, “The string-to-string correction problem,” *Journal of the ACM (JACM)*, vol. 21, no. 1, pp. 168–173, 1974.

[92] Z. Wang, C. Zang, J. A. Rosenfeld, D. E. Schones, A. Barski, S. Cuddapah, K. Cui, T.-Y. Roh, W. Peng, M. Q. Zhang *et al.*, “Combinatorial patterns of histone acetylations

and methylations in the human genome," *Nature genetics*, vol. 40, no. 7, pp. 897–903, 2008.

[93] W. G. Wee, "Generalized inverse approach to adaptive multiclass pattern classification," *IEEE Transactions on Computers*, vol. 100, no. 12, pp. 1157–1164, 1968.

[94] Y. Xiao, K. Mehrotra, C. Mohan, and R. Raina, "Utilizing cis-elements to refine gene regulatory networks," in *Bioinformatics and Biomedicine (BIBM), 2013 IEEE International Conference on*. IEEE, 2013, pp. 65–71.

[95] Y. Xiao, K. G. Mehrotra, D. G. Allis, and P. N. Borer, "A fast sorting algorithm for aptamer identification using deep sequencing," in *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*. IEEE, 2014, pp. 759–763.

[96] Y. Xiao, K. Mehrotra, C. Mohan, P. Choudhary, and R. Raina, *Prediction of biological functions by histone modification patterns profiling*. The International Society for Computers and Their Applications (ISCA), 2017, pp. 217–222.

[97] J. J. Yu, V. A. V. Smith, P. P. P. Wang, A. J. A. Hartemink, and E. D. E. Jarvis, "Advances to Bayesian network inference for generating causal networks from observational biological data." *Bioinformatics*, vol. 20, no. 18, pp. 3594–3603, Dec. 2004.

[98] H. Zheng, L. Zhang, C. Linghu, and J. Zhang, "Peak detection of histone modifications based on chip-seq data in human genome," in *2012 5th International Conference on BioMedical Engineering and Informatics*, Oct 2012, pp. 898–901.

VITA

NAME OF AUTHOR: Yiou Xiao

PLACE OF BIRTH: Urumqi, Xinjiang, China

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

Syracuse University, USA
Nanjing University, China

DEGREES AWARDED:

M.S., 2009-2011, Syracuse University, USA
B.E., 2005-2009, Nanjing University, China

PROFESSIONAL EXPERIENCE:

Jun-Sep, 2015	Database Engine Developer @Futurewei Tech, Inc. Santa Clara, CA • Developed plan tree visualization toolkit plugin for PostgreSql for easy inspection and debugging. – Python, matplotlib • Improved the SMP concurrent processing module by refactoring process based to inter-thread memory share for efficiency. – C

May-Sep 2012	*Research Assistant / Software Developer @Chemistry Dept., Syracuse University* Developed an efficient nucleotide sequences indexing and counting algorithm with neighborhood matching. – C
Summer 2007	J2EE developer, Gloriscience Tech. Shanghai, China Refactored the legacy indoor navigation system using SVG compatible graphs for Yiwu retail market. – Java, EJB

PUBLICATIONS:

1. Prediction of biological functions by histone modification patterns profiling (Y. Xiao, P. Choudhary, K. G. Mehrotra, C. K. Mohan, and R. Raina), in Proc. 9th International Conference on Bioinformatics and Computational Biology (BICOB), March 2017.
2. Efficient Classification of Binary Data Stream with Concept Drifting Using Conjunction Rule Based Boolean Classifier (Y. Xiao, K. G. Mehrotra, and C. K. Mohan), in Proc. IEA-AEI 15, Springer, M. Ali et al. (Eds.): LNAI 9101, pp. 456-457, 2015.
3. A fast sorting algorithm for aptamer identification using deep sequencing Y. Xiao, K. G. Mehrotra, D. G. Allis and P. N. Borer, "A fast sorting algorithm for aptamer identification using deep sequencing," 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014), Beijing, 2014, pp. 759-763.
4. Utilizing Cis-elements to Refine Gene Regulatory Network (Y. Xiao, K. Mehrotra, C. Mohan, and R. Raina), in Proc. IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Shanghai (China),Âă pp. 65-71, Dec. 2013.
5. Acyclic Identification of Aptamer from Over-Represented Libraries Using Hash

Functions (Y. Xiao, K. Mehrotra, C. Mohan, and P.N. Borer), in Proc. 39th IEEE Northeast Bioengineering Conference (NEBEC), 2013.

REWARDS:

2017	Syracuse University Poster Competition: Best Poster award in EECS
2017	Syracuse University Research Pitch Competition: 3rd place in Engineering School.
2016	Syracuse TeHack hackthon top prize (smart cat litter box)
2012	Syracuse University Fellow Award (2012- 2016)
2011	Outstanding Graduate Student in CS (GPA top 1)
2010	Scholarship for academic excellence, EECS Syracuse University
2008	Ren Min Scholarship in Nanjing University