1-1-2012

Differential modeling for cancer microarray data

Omar Odibat
Wayne State University,

DIFFERENTIAL MODELING FOR CANCER MICROARRAY DATA

by

OMAR ODIBAT

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2012

MAJOR: COMPUTER SCIENCE

Approved by:

_____

Advisor Date

_____

_____

_____

DEDICATION

To the loving memory of my father

To my beloved mother, for her prayers to me

To my wife, who has supported me in all my endeavors

To my beautiful daughters

Sarah & Hala

ACKNOWLEDGMENTS

All thanks and praise to Allah.

I would like to thank my advisor Dr. Chandan Reddy for being a great advisor. His ideas and tremendous support had a major influence on this thesis. His support, guidance, advice throughout the research project, as well as his pain-staking effort in proof reading the drafts, are greatly appreciated. Indeed, without his guidance, I would not be able to put the topic together. Special thanks to my committee, Dr. Mohammad Ramzi, Dr. Shiyong Lu and Dr. Dongxiao Zhu for their support, guidance and helpful suggestions. Their guidance has served me well and I owe them my heartfelt appreciation.

A penultimate thank-you goes to my wonderful parents. Their love provided my inspiration. I owe my mom everything and wish I could show her just how much I love and appreciate her. I thank my brothers and sisters for encouraging and supporting me during my study. I would like to express my special and best thanks for my parents in-law for their unconditional support.

My friends in US, Canada, Jordan and other parts of the world were sources of joy and support. I am very happy that, in many cases, my friendships with you have extended well beyond our shared time in Detroit.

My final, and most heartfelt, acknowledgment must go to my wife Dr. Noor Alaydie , whose love and support allowed me to finish this journey. Her support, encouragement, and companionship has turned my journey through graduate school into a pleasure. For all that, and for being everything I am not, she has my everlasting love. Many special thanks go to my beautiful daughters Sarah and Hala.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation

Microarray studies are used to measure the expression level of thousands of genes under various conditions in different cells [98, 125]. Capturing the changes between two biological phenotypes is a crucial task in understanding the mechanisms of various diseases. Differential analysis methods, such as differential expression analysis and differential network analysis, are useful in understanding the biological processes induced by the conditional changes. The existing approaches depend on individually testing the changes in the expression level of each gene. However, it was shown that disease candidate genes are not marked only by the changes in their expression levels, but also by the changes in the gene-gene correlation and the changes in the network structure [90]. We propose two computational methods to find these differential genes. Three types of differential (or discriminative) genes are being identified using the proposed work: differential hubs, differential subnetworks and differential co-clusters.

1.2 Cancer Microarray Data and Phenotypic Variations

Recent advances in DNA microarray technologies have revolutionized the analysis of genes and proteins, and have made it possible to simultaneously measure the expression levels of thousands of genes. The expression level of a gene is a measurement of the frequency of the gene expression, and it can be used to estimate the current amount of the protein in a cell the gene codes for [70]. Usually, the number of genes is significantly larger than the number of biological samples, and it becomes crucial to identify subsets of genes that are relevant to the biological problem under study.

The gene expression data can be organized in two-dimensional matrices where the rows represent genes, and the columns represent various possible phenotypes such as normal cells,

cancerous cells, drug treated cells or time series points. There are various kinds of phenotypic variations. Examples of such phenotypic variations include the following:

- Different tissue types: e.g., normal VS. cancerous [3, 53].
- Different class sub-types: e.g., acute lymphoblastic leukemia (ALL) VS. acute myeloid leukemia samples (AML) [48].
- Different stages of cancer: early stage VS. developed stage of prostate cancer [99].
- Different subject type: e.g., male VS. female [131].
- Different group types (racial disparity): African-American VS. Caucasian American [71, 69].
- Different time points [47].
- Different organisms [65, 101].

In each of these examples, there are two classes of biological samples. We refer to them as class A and class B. Each class has the same set of genes, but the gene expression values and their activities are different between the two classes. Differential analysis methods, such as differential expression analysis and differential network analysis, are useful in understanding the biological processes induced by the conditional changes [26]. *The goal of the differential analysis of gene expression data is to identify the set of differential genes that are responsible for the differences between two classes of biological samples*. Most of the existing computational approaches depend on testing the changes in the expression levels of the each single gene individually. In this work, we propose novel computational methods approaches to efficiently identify the differential genes.

1.3 Main Challenges

Identifying the differential genes from the gene expression data is a challenging task due to the following issues:

- **Incorporating the class-labels**. Differential analysis methods are used to extract patterns that are highly correlated in one class compared to the other class. To identify these class-specific patterns, it is crucial to effectively incorporate the class labels of the samples to analyze the gene expression data [32].

- **Types of changes**. It was shown that the differential genes are not marked only by the changes in their expression levels, but also by the changes in the gene-gene correlations and the changes in the network structures.

- **Pattern-based analysis**. The activities of the genes are not independent of each other. Thus, it becomes critical to be able to study groups of genes in the context of differential analysis rather than analyzing single genes one at a time.

- **Heterogeneous samples (or cancer subtypes)**. Due to the heterogeneity of the samples or the existence of cancer subclasses, a subset of genes can be correlated in any subset of the samples. Hence, it is important to develop computational algorithms that can capture such differential subspace patterns. We refer to these patterns as biclusters or co-clusters.

- **Noisy cancer microarray data**. The expression data contains a huge amount of noise [68]. Hence, the differential analysis algorithms should be robust against noise.

- **Overlapping-patterns**. A gene can be involved in several biological pathways. Hence, the same gene can belong to more than one group [95, 34].

- **Positive and negative correlations**. There are different types of correlations between genes in any cell. Examples of such relationships are positive and negative correlations. In a positive correlation, genes show similar patterns while in a negative correlation,

genes show opposite patterns. Since it is possible that genes with both types of correlations exist in the same biological pathway [67], there is a need for a computational model that captures both types of correlations simultaneously [143].

The existing methods do not handle all of the above challenges. The proposed work tackles all of the above challenges. Specifically, the differential networking and differential co-clustering algorithms effectively incorporate the class labels of the biological samples in the search process, and they can identify groups of differential genes (differential subnetworks and differential co-clusters). Moreover, both approaches allow the discovery of overlapping patterns that contain negative and positive correlations. Furthermore, these approach are robust against noise.

The proposed differential network approach can analyze the changes in the network structure and identify differentially connected genes in the form of differential hubs and differential subnetworks.

Co-clustering can be used to simultaneously cluster both dimensions of a data matrix by utilizing the relationship between the two entities [113], and it helps in discovering local patterns that cannot be identified by the traditional one-way clustering algorithms. The proposed differential co-clustering method can identify differential subspace patterns. Therefore, it can handle heterogeneous samples (or cancer subtypes).

1.4 Our Contributions

In this work, we propose to develop novel computational methods to find the differential genes between two phenotypes. The proposed approaches are: differential network analysis and differential co-clustering. The proposed models can quantitatively and qualitatively characterize the differences between two classes (or two phenotypes) and can provide better insights and understanding of various diseases. Figure 1.1 illustrates the overview of the proposed framework.

As shown in this Figure, the input to the proposed work is a dataset that consists of two phe-

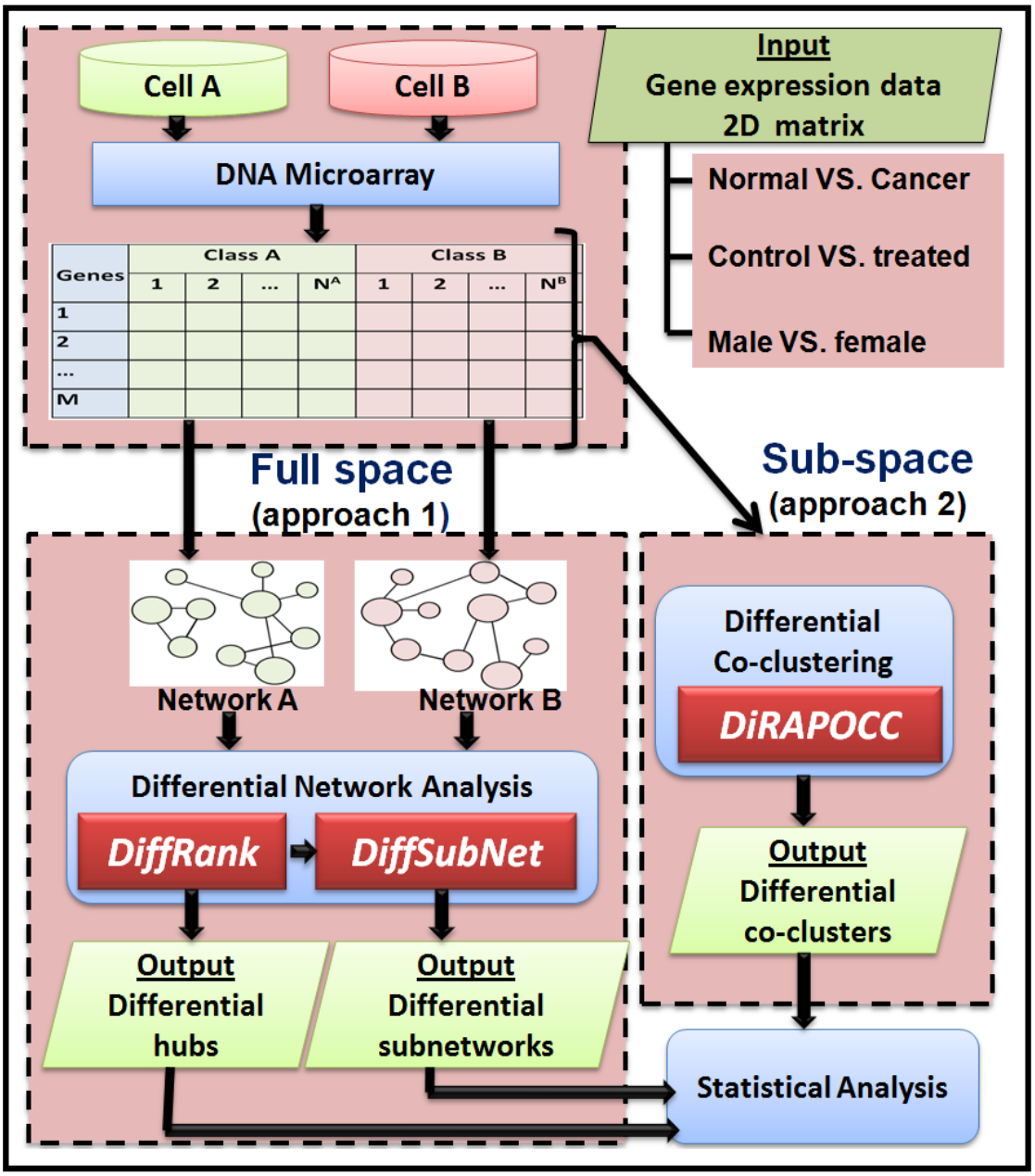


Figure 1.1: Overview of the proposed approach 1 and approach 2.

notypes (or two classes) of biological samples. This dataset is organized as a two-dimensional matrix. Each row in this matrix represents a single gene, and each column represents a biological sample which belongs to one of the two classes. The goal is to identify the set of differential genes between the two phenotypes. We propose two computational methods to find these dif-

ferential genes. Three types of differential genes are being identified using the proposed work: differential hubs, differential subnetworks differential co-clusters. The output of the proposed work will be statistically analyzed and the corresponding p-values are computed. It is worth mentioning that the proposed work can be applied on other problem that has two classes of samples. The purpose of this thesis is three-fold described as follows:

1. **A review of the differential analysis methods.** We review the state-of-the-art approaches for differential analysis of gene expression data including the following categories: differential expression, differential variability, differential co-expression, differential biclustering and differential networking methods. We characterize each category, and we observe certain relationships between them.

2. **A novel differential network analysis method.** We propose novel differential network analysis methods that is composed of two algorithms, namely *DiffRank* and *DiffSubNet*, to identify differential hubs and differential subnetworks, respectively. In this approach, two datasets are used to construct two networks, and then the problem of identifying differential genes is transformed to the problem of comparing two networks to identify the most differential network components.

3. **A novel differential co-clustering method.** We propose a novel differential co-clustering approach to efficiently identify discriminative co-clusters from large datasets. To achieve this goal, we propose two novel algorithms. The first algorithm is a novel co-clustering algorithm: **R**anking-based **A**rbitrarily **P**ositioned **O**verlapping **C**o-Clustering (*RAPOCC*), which can be used to efficiently find arbitrarily positioned co-clusters in the data matrix. This algorithm is then extended to discover discriminative co-clusters: **Di**scriminative **RAPOCC** (**Di-RAPOCC**) by incorporating the class information into the co-cluster discovery process to extract class-specific co-clusters.

The proposed novel differential network analysis is composed of two algorithms (*DiffRank*

and *DiffSubNet*) which can be used to identify differential hubs and differential subnetworks, respectively. In this approach, two datasets are represented as two networks, and the problem of identifying differential genes is transformed to the problem of comparing two networks to identify the most differential network components [96]. Studying such networks can provide valuable knowledge about the data. The *DiffRank* algorithm ranks the nodes of two networks based on their differential behavior using two novel differential measures: differential connectivity and differential betweenness centrality for each node [97]. These measures are propagated through the network and are optimized to capture the local and global structural changes between two networks [98]. Then, we integrate the results of this algorithm in the second proposed differential subnetwork algorithm (*DiffSubNet*). This algorithm aims to identify sets of differentially connected genes. We demonstrated the effectiveness of these algorithms on synthetic datasets and real-world applications and showed that these algorithms are capable in identifying meaningful and valuable information compared to some of the baseline methods that can be used for such a task.

The goal of the differential co-clustering approach is to discover a distinguishing set of gene patterns that are highly correlated in a subset of the samples in one phenotype but not in the other [99]. Due to the heterogeneity of some diseases such as cancer, the set of genes can be co-expressed only in a subset of the samples (subspace co-expressions). Hence, the proposed differential co-clustering approach does not require correlated genes to be similar under all the features (biological samples). To achieve this goal, we propose a novel co-clustering algorithm, **R**anking-based **A**rbitrarily **P**ositioned **O**verlapping **C**o-Clustering (*RAPOCC*), to efficiently extract significant co-clusters. This algorithm optimizes a novel ranking-based objective function to find arbitrarily positioned co-clusters, and it can extract large and overlapping co-clusters containing both positively and negatively correlated rows [95]. Then, we extend this algorithm to discover discriminative co-clusters by incorporating the class information into the co-cluster search process. The novel differential co-clustering algorithm, called **Di**fferential

RAPOCC (*Di-RAPOCC*), can efficiently extract the discriminative co-clusters from labeled datasets. We also characterize the discriminative co-clusters and propose three novel measures that can be used to evaluate the performance of any differential subspace algorithm.

1.5 Organization of this Thesis

This thesis is composed of the following three major parts:

- **Part 1: Review of existing methods (Chapter** 2). In this Chapter, we review the state-of-the-art approaches for differential analysis of expression data. We also discuss the main limitations and problems in the existing approaches and explain how the proposed approaches solve these problems.

- **Part 2: Differential networking approach (Chapters** 3 **and** 4). In chapter 3 we present the proposed differential hubs ranking algorithm (*DiffRank*), and in Chapter 4 we present the proposed differential subnetwork detection algorithm (*DiffSubNet*). We present the results of each algorithm on synthetic and real datasets.

- **Part 3: Differential co-clustering approach (Chapters** 5 **and** 6). In chapter 5 we present the proposed co-clustering algorithm (*RAPOCC*), and in Chapter 6 we present the proposed differential co-clustering algorithm (*Di-RAPOCC*). We present the results of each algorithm on synthetic and real datasets in the corresponding chapter along with the comparisons with other algorithms available in the literature.

Finally, we summarize and conclude our work and provide some possible directions for future work in chapter 7.

CHAPTER 2

A REVIEW OF DIFFERENTIAL ANALYSIS ALGORITHMS

2.1 Overview

Microarray studies are used to measure the expression level of thousands of genes under different conditions in different cells [125]. These cells have the same set of genes, but the gene expression levels and their activities are different. There are several examples of such phenotypic variations [115] such as: different tissue types: e.g., normal vs cancerous [3, 53], or different class types: e.g., acute lymphoblastic leukemia (ALL) vs acute myeloid leukemia samples (AML) [48]. In these examples, the expression levels of the same genes are measured under two classes of conditions.

Capturing the changes between two biological conditions, such as normal versus cancer, is a crucial task in understanding the causes of diseases. Differential analysis methods, such as differential expression analysis and differential co-expression analysis, are helpful in understanding the biological processes induced by the conditional changes. In this chapter, we survey the state-of-the-art approaches for differential analysis of gene expression data including the following categories: *differential expression*, *differential variability*, *differential co-expression* (or *differential clustering*), *differential biclustering* (or *differential co-clustering*) and *differential networking* methods. These methods are classified in Figure 2.1 and summarized in Table 2.1. We characterize each category, and we make some observations about the relationships between them.

Basically, the differential analysis methods can be classified into three main categories. The first category is single gene analysis in which each gene is tested individually to identify *differentially expressed genes*. The methods in the second category identifies differential patterns by

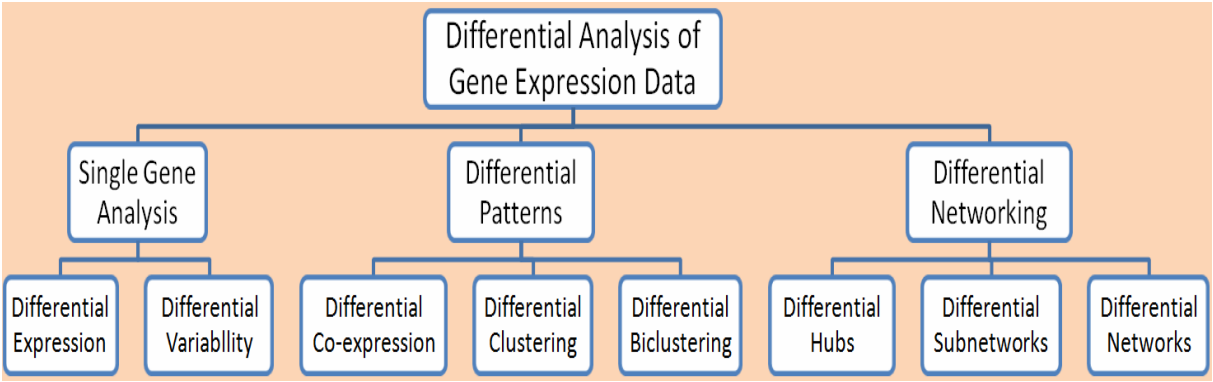


Figure 2.1: Classification of the state of the are differential analysis methods of gene expression data.

testing the changes in the gene-gene correlation to identify *differentially co-expressed genes*. The third category is the differential networking approach. In this approach, a network is constructed from the expression data of each phenotype, and then the two networks are analyzed to identify *differentially connected genes*.

2.2 Single Gene Analysis

2.2.1 Differential Expression (DE)

Several methods have been used to identify differentially expressed (DE) genes that are related to a certain phenotype [141]. The differentially expressed genes can be identified by testing the statistical significance of the changes in the mean level of the expression level of each individual gene. A threshold level is defined on the test statistics, usually the t-test or fold change [51], and a correction method, such as FDR, is used to adjust for the multiple hypothesis testing problem [129, 140]. The DE methods are helpful only when the disease genes are differentially expressed. However, there are some cases where the disease genes are not differentially expressed such as in mutations and post-translational modifications of a gene product. In these cases, the function of the gene is affected but not its expression level [32]. Therefore, depending only on the change in the mean of the expression level of the genes can not identify all the disease genes [32].

2.2.2 Differential Variability (DV)

Differential variability (DV) was proposed to identify genes with a significant change in the variance of expression between two conditions [104]. In this type of analysis, ANOVA (ANalysis Of VAriance) [105] or the F-test [55] can be used to identify the DV genes. Both of the differential expression and the differential variability depend on statistically testing each gene individually and do not capture the relationships between the genes. Since the activities of the genes are not independent of each other, there is a critical need to study groups of genes rather than performing a single gene analysis.

2.3 Differential Patterns

2.3.1 Differential Co-expression (DC)

Functionally related genes usually exhibit expression patterns (correlated expression profiles) [122, 81, 27]. Differentially Co-expressed (DC) methods aim to find the differences in the co-expression patterns in normal and disease samples [32]. It was shown that some disease genes were highly differentially co-expressed but not differently expressed. In addition, differential expression does not necessarily indicate biological significance [58]. Differentially co-expressed (DC) genes are correlated in one type of samples but not in the other [136, 26]. The co-expression relationships (or correlation) can be measured by several functions, such as Pearson correlation coefficient, and they reflect functional relationships. Since genes are not independent and they interact with each other, the differential co-expression methods consider the relationships between different genes, while DE and DV methods are based on testing each gene individually [58].

To achieve a similar goal, differential Clustering Analysis (DCA) was proposed in [65] to find differentially correlated groups of genes between two conditions. This method was used to identify conserved and diverged co-expression patterns when comparing two organisms.

2.3.2 Differential Biclustering (DB)

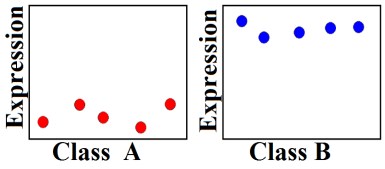
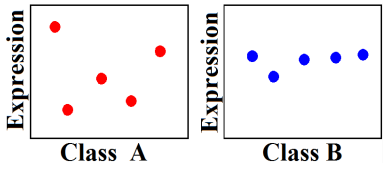
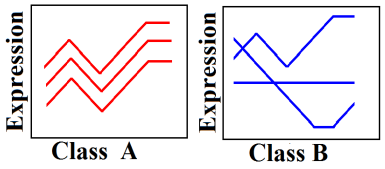
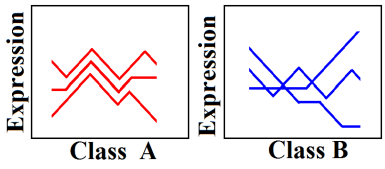
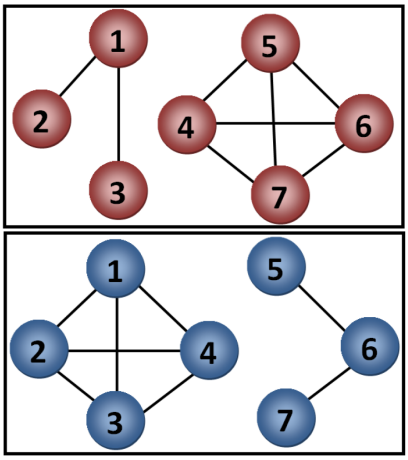
A bicluster (or co-cluster) is a subset of co-expressed genes under a subset of samples [95]. Differential Biclustering is used to extract differential biclusters from the gene expression data where the samples belong to one of the two classes. Since some genes are activate only in a subset of the samples, identifying the genes that are over- or down-expressed in some but not all the samples in a group is very important [139]. The genes in the differential biclusters have strong correlation in one class but not in the other, or they may have different types of co-expression among the two classes [99]. Differential biclustering is useful when the biological samples are assumed to be heterogenous or have multiple subtypes. The main difference between differential co-expression and differential biclustering is that in the first approach the correlation between any two genes is computed based on all the samples, while in the second approach, two genes can be similar in a subset of the samples.

2.4 Differential Network Analysis

Networks have been extensively used to model the gene activities and their interactions [62, 8, 125, 50, 132, 30, 29, 93]. These networks consist of genes as the nodes and the interactions between them as the edges. Studying the topology and the functionality of these networks can provide valuable knowledge for understanding the roles of genes in several diseases [32]. Differential analysis of networks has led to important results in studying the phenotypic differences across different conditions [56, 18, 17, 142, 131, 119, 5, 54, 134, 87, 94]. The set of genes which cause network topological changes may serve as biomarkers [145]. In addition, network comparison can be used to provide insights into disease-specific alterations [32] and to examine the effects of a certain treatment [145]. The main challenge in the differential network analysis is to identify the important differences between two networks.

There are some differential networking methods that have been proposed in the literature. We categorize these methods into three basic categories: node-level (hubs), subnetwork level, and network level.

Table 2.1: Differential analysis methods of gene expression data.

Type		Definition	Illustration	Examples
Differential Expression (DE)		*Testing the changes in the* **mean** *level of expression of each gene.*		SAM [129], Pattern analysis [10], Samroc [14], PUL [130], Maximum-Likelihood [63], B-statistics [88].
Differential Variability (DV)		*Testing the changes in the* **variance** *of* expression of *each gene.*		ANOVA [105], DV [55], AlteredExpression [104], Kerr *et al.* [72], Variance ratio [23]
Differential Co-expression (DC)		*Testing the changes in the co-expression* **patterns** *of genes in* **all** *of the samples.*		ECF-statistic [75], CoXpress [136], DCIM [43] MIClique [147], GSCA [28], DGCL [82] dCoxS [26], (Kostka and Spang 2004) [74]
Differential Biclustering (DB)		*Testing the changes in the co-expression* **patterns** *of genes in* **a subset** *of the samples.*		BiModule [100], SDC [39], DiBiCLUS [99]. FDCluster [135] DeBi [112]
Differential Networking	**Differential Hubs**	*Testing the changes in the* **connectivity** *of* **single gene**.		Differential Connectivity [109] Differential Hubbing[60] MDA-*single gene* [47] DiffK [44], DiffRank [97].
	Differential Subnetworks	*Testing the changes in the* **connectivity** *of* **groups of genes**.		Differential clique [132], DiffCoEx [127], COSINE [90], PNA [73], Liu *et al.*, [87], OptDis [31], DifferentialNW [17], DDN [146], postOR [30], jActiveModules [62], MDA-*class of genes* [47].
	Differential Networks	*Testing the changes in the* **structure** *of the* **networks**		Degree distribution [108], Degree similarity [144], Network diameter [144], MDA-*modular structures* [47].

2.4.1 Differential Hubs (DH)

The goal is to identify the differentially connected genes (or differential hubs). Although this type of analysis focuses on identifying single genes as differential hubs, the correlation between each gene and each other gene is considered rather than testing each gene individually as the DE and DV methods. To compare the genes between two gene networks, several differen-

tial measures such as differential connectivity have been defined in [18, 109, 44, 38, 126, 60]. Some methods are based on performing permutations and statistical test such as the MDA test [47]. Most of these methods depend on pair-wise comparisons of the genes based on their degrees. Therefore, as will be discussed in the next chapters, we propose an efficient algorithm to capture all the local and global changes between two networks.

2.4.2 Differential Subnetworks (DS)

In this category, the goal is to identify differentially connected groups of genes among two co-expression networks. There are a few differential network analysis methods that have been proposed to identify differential subnetworks when comparing two biological conditions. Most of the existing methods merely perform pairwise comparisons based on: (i) the nodes (jActiveModules [62], DDN [145] and OptDis [31]) or (ii) the edges (DifferentialNW [17], Differential clique analysis [132], DiffCoEx [127], postOR [30] and [87]) or (iii) both of the nodes and the edges (COSINE [90] and PNA [73]).

Some of the recent methods, such as OptDis [31] and CRANE [29], depend on integrating protein-protein interaction (PPI) data to define the networks, and they use the gene expression data to measure the changes of the expression levels of the genes between two biological conditions. OptDis [31] uses a color coding algorithm to find the subnetworks. CRANE [29] works on binary gene expression data, the digitization which is sensitive to several user-defined parameters. COSINE [90] is a recent method that uses the F-statistic to measure the differential expression of each gene, and it uses the Expected Conditional F-statistic (ECF-statistic) to measure the differential genegene co-expression across different groups. Then, a genetic algorithm is used to search for the highest scoring subnetwork. Differential clique analysis was defined in [132]. In this approach, clique membership is combined with differential correlation. DiffCoEx [127] works based on the WGCNA model [76]. This method uses a new dissimilarity measure computed from the topological overlap that is found using Pearson correlation. postOR [30] compares the posterior probabilities of connectivity for each gene pair

across two disease states, expressed as a posterior odds-ratio (postOR) for each pair, which is then used to compute the overall differential connectivity for each gene sets. There are some statistical based methods to identify differentially expressed set of genes from gene networks. Such methods include: MDA [47] and DDN [146].

2.4.3 Differential Networks(DN)

In this category, the goal is to test if the given connectivity of the overall two gene co-expression networks are different or not. In [108], the degree distribution of each network was used to compare the two gene networks, and in [47] a statistical test was defined to test the changes in the overall modular structures of the two networks. However, several other network features can be used to compare two gene co-expression networks, such as the average shortest paths length, the network diameter, the mean clustering coefficient and the degree similarity [144].

2.5 Other Related Topics

- **Gene set enrichment** Incorporating priori knowledge has been used in several methods to identify significant genes, gene sets or pathways [12]. Gene set enrichment analysis aims to identify differentially expressed groups of genes [123, 37, 1]. This types of analysis primarily depends on prior knowledge about the groups of genes processes [86]. Each group consists of functionally related genes such as certain pathways or biological processes [117]. Examples of such gene set enrichment methods include GSEA [123], SAFE [9] and GNEA [85]. A review of such methods can be found in [1].

- **Over-Representaion Analysis (ORA)** Over-Representaion Analysis tests whether a given gene set, such as Gene Ontology (GO) terms, is statistically over-represented in a list of DE genes based on the hypergeometric test [148].

2.6 Limitations of the Existing Work

The differential analysis methods in the single gene analysis category do not capture the correlations between genes. The differential expression and the differential variability methods depend on statistically testing each gene individually. Since the activities of the genes are not independent of each other, there is a critical need to study groups of genes rather than performing a single gene analysis. To capture the correlations between genes, co-expression or clustering methods can be used to identify gene patterns.

Differential co-expression and differential clustering methods have been used to find differentially correlated groups of genes between two phenotypes and to identify class-specific patterns. These methods use the entire feature space to find the differential genes for each phenotype. However, these genes can be correlated only in a subset of the cancerous samples due to the heterogeneity in the sample space [95]. Hence, it is important to develop a model that can identify discriminative patterns that are correlated in a subset of the the feature space. Co-clustering has been proposed to capture the patterns that are correlated in a subset of features, but it cannot handle discriminative patterns in labeled datasets. In this work, the author proposes a novel algorithm (*Di-RAPOCC*) to discover discriminative co-clusters by effectively incorporating the class information into the co-cluster search process. The proposed algorithm captures large and overlapping differential co-clusters that contain positive and negative correlations. In addition, the proposed algorithm is robust against noise.

In the context of differential network analysis, there are a few differential measures that have been proposed to identify the differential hub genes. However, these methods depend on pair-wise comparisons of the genes based on their degrees. Therefore, the author proposes an efficient algorithm to capture all the local and global changes between two networks. our proposed *DiffRank* algorithm ranks the genes based on their differential behavior using two novel differential measures, namely, differential connectivity and differential betweenness centrality.

Compared to identifying differential hubs, identifying differential subnetworks is even

more challenging since it optimizes for a group of connected nodes that are specific to one particular class. Most of the existing methods merely perform pairwise comparisons based on the nodes[62] or the edges [17, 132] or both of the nodes and edges [90, 73]. Hence, these methods do not capture the global changes in the network because they focus only on the local comparisons. Here, the author proposes a novel algorithm (*DiffSubNet*) to identify the differential subnetworks. This algorithm incorporates the differential node scores obtained from the *DiffRank* algorithm. The differential subnetworks are groups of strongly connected nodes (dense subnetworks) in one network but not in the other network. These subnetworks can overlap within the same network, but they should not overlap between the two networks.

CHAPTER 3

RANKING DIFFERENTIAL HUB GENES

3.1 Motivation

Networks have been extensively used to model various complex systems such as online social networks, co-authorship and biological networks. These networks consist of data objects as the nodes and the interactions between them as the edges. Studying such networks can provide valuable knowledge about the data objects and their interactions. The interactions between the data objects depend on the domain in which these data objects are studied.

Normal and cancerous cells have the same set of genes, but some of these genes are differentially wired in the cancerous cells, which results in two different gene interaction networks [32]. Here, the nodes are the genes, and the edges represent the interactions between the genes. Since the genes that have strongly altered connectivity play an important role in the disease phenotype [32], finding the differential genes can be used in several applications such as identifying disease-causing genes and examining the effects of a certain treatment [32].

3.1.1 Differential Gene Network Analysis

Gene networks have emerged as an efficient tool in modeling gene activities and in understanding the roles of genes in several diseases [32]. The main advantage of differential networking over the other methods, is that using networks will enable studying the whole spectrum of pair-wise relationships [38]. Differential analysis of networks has led to important results in studying the phenotypic differences across different conditions [44], identifying disease-causing genes and in examining the effects of a certain treatment [32]. Moreover, the set of genes which cause network topological changes may serve as biomarkers [145]. However, network comparison is a challenging problem, and it was shown that it is an NP-complete problem [17, 106].

The goal of differential network analysis is to identify the differentially connected genes (or differential hubs). Although this type of analysis focuses on identifying single genes as differential hubs, the correlation between each gene and with the other genes is considered rather than testing each gene individually as in the differential expression (DE) [129] and the differential variability (DV) [55] methods. Both of the DE and the DV methods depend on statistically testing each gene individually using the T-test and the F-test respectively. Therefore, these methods do not capture the relationships between the genes. To overcome these problems, networks have been successfully used to model the gene activities and their interactions. These networks consist of genes as the nodes and the interactions between them as the edges. Studying the topology and functionality of these networks can provide valuable knowledge for understanding the roles of genes in several diseases [32].

The main technical challenge of exploiting the network structure to find the differential hubs is to find all the differences between two networks. A straightforward solution is to transfer this problem to solving the subgraph isomorphism problem. Unfortunately, this is not desirable as it is computationally infeasible, and it was shown that solving the subgraph isomorphism problem is NP-complete problem [106].

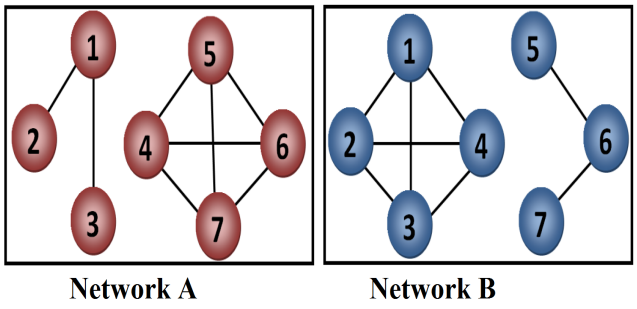


Figure 3.1: A simple illustration of differential hubs.

3.1.2 Related Work

In the biological domain, there are some differential measures that have been proposed to measure the differences between two gene networks. To compare the genes between two gene

networks, several differential measures such as differential connectivity have been defined in [109, 44, 60], some methods are based on performing permutations and statistical test such as the MDA test [47]. However, most of these methods depend on pair-wise comparisons of the genes based on their degrees. Therefore, we propose an efficient algorithm to capture all the local and global changes between two networks.

Toy Example: As an example, two networks are shown in Figure 3.1. In this example, it can be seen that the gene 4 should be identified as the differential gene when comparing network A and network B. However, this gene has the same degree (which is 3) in both networks. Therefore, depending only on comparing the degree of each gene cannot capture all the differences between two gene networks. Using the proposed method, gene 4 will be the top ranked differential gene in this figure.

Our goal is to identify the differential hubs by analyzing two interaction networks. We combine differential network analysis with ranking in one framework and propose a novel ranking algorithm, *DiffRank*, which ranks the nodes of two networks based on their differential behavior in the two networks. To achieve this goal, we define novel measures such as differential connectivity and differential centrality for each node. These measures are propagated through the network and are optimized to capture the changes in the local and global structures between two networks.

3.1.3 Our Contributions

The main contributions of this chapter can be summarized as:

1. We propose *DiffRank* algorithm to rank the hubs of two networks based on their differential behavior in the two networks and to identify the differential hubs.

2. We propose two novel differential measures:

   (a) A local structure measure, *differential connectivity*, to capture the local differences between two networks based on their weighted edges.

(b) A global structure measure, *differential betweenness centrality*, to capture the global differences between two networks based on the shortest paths

3. We develop a simulator for generating synthetic differential scale-free networks based on two models to evaluate the proposed algorithm.

The proposed algorithm has two salient features. First, it can effectively capture the differences in both local and global structures between two networks. Second, it iteratively propagate the novel differential scores through the network until convergence to obtain accurate rankings for all the nodes. We show that *DiffRank* is motivated by and well reflects the existing observations about the differences between two networks. Empirical experiments on three different applications show that our approach is effective and outperforms various baselines. To the best of our knowledge when this thesis was written, *DiffRank* is the first algorithm to rank the nodes of two networks based on their differential behavior and to identify the differential hubs.

3.2 Preliminaries and Problem Formulation

We will now introduce the notations to be used in the rest of the chapter; then, we formally present the problem statement.Given two gene networks, represented by graphs $G^A(V, E^A)$ and $G^B(V, E^B)$, where V is the set of N nodes and E^c is the set of edges in G^c, $c \in \{A, B\}$. An edge between two genes u and v, with a weight $w^c(u, v)$ in G^c, determines the strength of the interaction between the genes. The weight of each edge must be a non-negative value, 0 if the nodes are not connected to each other, or 1 in unweighted graphs. We denote the degree of gene v in network c as k_v^c. The proposed algorithm can be applied on both directed and undirected networks. In this work, we focus our discussion to undirected networks with no self-links.

Problem Formulation: *Given two networks, G^A and G^B, the goal is to find the differential hubs that best explain the differences between the two networks. The final output of the*

DiffRank algorithm is a vector

$$\Pi =< \pi_1, \pi_2, ..., \pi_N >$$

where π_v denotes the rank of the differential node v.

A reasonable and accurate model for differential networks should not only capture the changes in the local structure, but also the changes in the global structure. Before formally introducing the algorithm, we first explain several key observations that motivate our approach.

Connectivity: The connectivity, or the degree, of a node is the number of other nodes that it is connected to. Nodes with the highest number of edges, known as the hubs, play an essential role in the analysis of networks. Pair-wise comparisons of the degree of each node in the two networks, as proposed in [44], may not lead to accurately identifying the differential hubs. For example, node 4 in Figure 3.1 has the same degree in both networks but the edges are different.

Centrality: Centrality is important in understanding many networks such as social networks [20], co-authorship networks [36] and biological networks [49]. Moreover, central nodes can have high influence on their neighbors [137]. Betweenness Centrality (BC) can be used to measure the centrality for each node, which is proportional to the sum of the shortest paths passing through it [42].

Identifying the shortest paths between two nodes is critical in several applications, such as social and biological networks [49], and the influence maximization problem [21]. Usually, the weights of the edges represent the strength of the interactions (or correlations) between the nodes. Therefore, distance values should be calculated from the weight values in order to calculate the shortest paths. For example, if $w(u, v)$ is the weight of interactions between two nodes u and v, then the weight on each edge can be translated to distance path using $1-w(u, v)$ or $-log(w(u, v))$ [21]. We expect these intuitions and observations to be helpful in designing the proposed algorithm.

3.3 The Proposed DiffRank Algorithm

The proposed model is composed of two measures: *differential connectivity* and *differential betweenness centrality*. These measures are optimized to capture the changes in the local structure and the changes in the global structure between two the networks respectively.

3.3.1 Differential Connectivity

Genes with the highest number of edges, known as hubs, play central roles in the analysis of networks. Differential connectivity measures the local differences between two networks, G^A and G^B, by considering the actual weights of all the edges, and it is defined as follows:

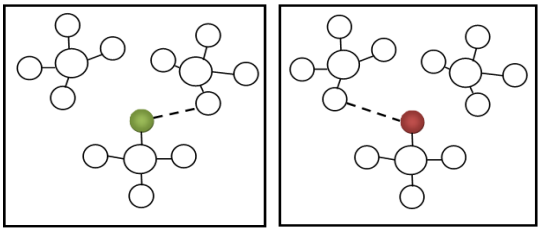
$$\Delta C^i(v) = \sum_{u=1}^{N} \frac{|w^A(u,v) - w^B(u,v)| \cdot \pi_u^i}{\sum_{z=1}^{N} |w^A(u,z) - w^B(u,z)|} \tag{3.1}$$

where π_v^i is the differential scores (or rank) of node v at the i^{th} iteration. It is initialized to $\frac{1}{N}$ and will be updated in each iteration (it can also be used to incorporate prior knowledge). If a given gene has the same set of edges in both networks with the same weights, then the differential connectivity of that node will be 0. On the other hand, when a node has different sets of edges (such as gene 4 in Figure 3.1), it will get a high value for the differential connectivity. In addition to the number of edges and their weights, the differential connectivity of each gene also depends on the differential scores of the neighbors it is connected to. A gene will be assigned a higher score if it is connected to many differential genes. Given two genes, u and v, the propagation of the differential score from u to v depends on three factors:

1. The weight of the edge (u, v) in both networks, denoted by $|w^A(u,v) - w^B(u,v)|$.

2. The current score of the gene u, denoted by π_u^i.

3. The weights of all the edges connected to u, denoted by $\sum_{z=1}^{N} |w^A(u,z) - w^B(u,z)|$.



(a) Network A (b) Network B

Figure 3.2: A simple illustration for differential betweenness centrality.

3.3.2 Differential Centrality

Centrality is an important measure in understanding biological networks because it is difficult to detect the changes in the expression level of the central genes by single gene analysis. However, these changes could significantly alter the topology of the network [41]. Hence, we integrate the notion of gene centrality into the proposed algorithm.

Betweenness Centrality (BC) can be used to measure the centrality of each node, which is proportional to the sum of the shortest paths passing through it [42]. If P_{st} is the number of the shortest paths from node s to node t, where $s \neq t$, and $P_{st}(v)$ is the number of the shortest paths from s to t that pass through a node v, where $s \neq v$ and $t \neq v$, then the BC of the node v can be computed as $BC(v) = \sum_{s \neq t} \frac{P_{st}(v)}{P_{st}}$ [41]. In gene co-expression networks, the weights of the edges represent the correlation between the genes. Therefore, distance values should be calculated from the correlation values in order to calculate the shortest paths. For example, if $w(u, v)$ is the correlation between two genes, then the distance between the two genes could be computed as $1 - w(u, v)$.

Comparing the values of BC may not detect the topological changes. For example, the shaded gene in Figure 3.2 has the same value of BC (which is 6) in both networks. However, the shortest paths that pass through that gene are different. Therefore, we propose to consider the shortest paths in our method. Let SP_v^c be a binary $N \times N$ matrix, such that $SP_v^c(s, t) = 1$ if one of the shortest paths from s to t passes through the node v in network $c = \{A, B\}$, where $s \neq t$, and it is 0 otherwise. We define differential betweenness centrality of a node v as

follows:

$$\Delta BC(v) = \sum_{s=1}^{N} \sum_{t=1}^{N} |SP_v^A(s,t) - SP_v^B(s,t)| \tag{3.2}$$

3.3.3 The DiffRank Algorithm

We propose *DiffRank* algorithm which iteratively optimizes an objective function that is a linear combination of differential connectivity and differential betweenness centrality (parameterized by λ) within a PageRank-style framework [52], such that the rank of each node v is computed as follows:

$$\pi_v^i = (1-\lambda) \cdot \frac{\Delta BC(v)}{\sum_{u=1}^{N} \Delta BC(u)} + \lambda \cdot \Delta C^i(v) \tag{3.3}$$

The parameter λ controls the trade-off between differential connectivity and differential betweenness centrality. It can be assigned any value in the range $[0, 1]$. When $\lambda = 0$, the ranking depends only on the differential betweenness centrality, and when $\lambda = 1$, the ranking depends only on the differential connectivity. Any other value of λ combines both terms in the ranking. We set λ to 0.75 based on some of the preliminary experiments we performed. The integration of the ΔBC term into Equation (3.3) adds significant global topological information to the differential analysis of networks.

3.3.4 Condition-specific Analysis

It is important to find the genes that are differentially rewired in the cancer cells. For this purpose, we introduce a second version of the proposed algorithm based on the particular network of interest. To find the differential nodes in network B, the differential connectivity ($\Delta C'$) for each gene can be redefined as follows:

$$\Delta C'^i(v) = \sum_{u=1}^{N} \frac{max(w_B(u,v) - w_A(u,v), 0) \cdot \pi_u^i}{\sum_{z=1}^{N} max(w_B(u,z) - w_A(u,z), 0)} \tag{3.4}$$

This new definition excludes any edge in the network of interest if the corresponding edge in

the other network has a higher weight. Similarly, the new definition of differential betweenness centrality, $\Delta BC^{'}$, includes the unique shortest paths that are in the network of interest and excludes the unique shortest paths in the other network.

$$\Delta BC^{'}(v) = \sum_{s=1}^{N} \sum_{t=1}^{N} max(SP_B^v(s,t) - SP_A^v(s,t), 0) \tag{3.5}$$

The second version of *DiffRank* is modified as follows:

$$\pi_v^i = (1-\lambda) \cdot \frac{\Delta BC^{'}(v)}{\sum_{u=1}^{N} \Delta BC^{'}(u)} + \lambda \cdot \Delta C^{'i}(v) \tag{3.6}$$

These two versions of *DiffRank* can solve the following problems:

1. Find the top differential genes; this can be solved by the first version of *DiffRank*. In this version, we solve the phenotypic distinction problem.

2. Find condition-specific differential genes; this can be solved by the second version of *DiffRank*. In this type of analysis, we focus on the set of genes that are active in the cancer networks (identifying disease-causing genes).

3.3.5 Preservation and Convergence

To begin with, all the nodes are initialized to $\frac{1}{N}$ (uniform distribution), so that the sum of the rankings is 1 i.e., $\sum_{v=1}^{N} \pi_v^i = 1$. The rankings will be updated in each iteration. There is no need to normalize after each step since the sum of the rankings is preserved to unity.

Lemma 1. *The sum of the node ranks Π_Δ obtained by DiffRank is preserved to unity.*

Proof. Let us assume that the algorithm is at the iteration i and $\sum_{v=1}^{N} \pi_v^i = 1$. Now, we will

show that the sum of the rankings is preserved for the next iteration $(i+1)$:

$$\begin{aligned}
\sum_{v=1}^{N} \pi_v^{i+1} &= \sum_{v=1}^{N} \left(\frac{(1-\lambda).\Delta BC(v)}{\sum_{u=1}^{N} \Delta BC(u)} + \lambda. \sum_{u=1}^{N} \Delta DC^i(v) \right) \\
&= (1-\lambda). \left(\frac{\sum_{v=1}^{N} \Delta BC(v)}{\sum_{u=1}^{N} \Delta BC(u)} \right) + \lambda. \left(\sum_{v=1}^{N} \sum_{u=1}^{N} \frac{|w^A(u,v) - w^B(u,v)|.\pi_u^i}{\sum_{z=1}^{N} |w^A(u,z) - w^B(u,z)|} \right) \\
&= (1-\lambda) + \lambda. \left(\sum_{u=1}^{N} \pi_u^i \frac{\sum_{v=1}^{N} |w^A(u,v) - w^B(u,v)|}{\sum_{z=1}^{N} |w^A(u,z) - w^B(u,z)|} \right) \\
&= (1-\lambda) + \lambda. \sum_{u=1}^{N} \pi_u^i \\
&= (1-\lambda) + \lambda = 1
\end{aligned}$$

□

One issue that needs to be resolved is handling the sinks (or isolated nodes). These nodes will be assigned uniform weighted edges to each other node in the network in order to ensure the convergence of the *DiffRank* algorithm [77].

Theorem 1. *The result from the DiffRank model converges to a unique rank vector.*

Proof. Let us define $M^{N \times N}$ as a square matrix, such that

$$M_{uv} = \frac{|w^A(u,v) - w^B(u,v)|}{\sum_{z=1}^{N} |w^A(u,z) - w^B(u,z)|}$$

We replace all rows with zeros by $\frac{1}{N}$. Now, M is considered to be a stochastic matrix in which the sum of each row is 1: $\sum_{v=1}^{N} M_{uv} = 1$, $1 \leq u \leq N$. Let P denote a vector of length N, such that

$$P_v = \frac{\Delta BC(v)}{\sum_{u=1}^{N} \Delta BC(u)}$$

then we will have $\sum_{v=1}^{N} P_v = 1$. Finally, we define a new matrix $M^{'}$ as follows:

$$M^{'} = \lambda.M + (1-\lambda).P^T$$

The combination of the stochastic matrix M, and the vector P reduces the effect of the isolated nodes $\lambda \in [0, 1]$. Now, the rank vector Π_Δ can be computed by solving the following eigenvector problem:

$$\Pi_\Delta{}^T M' = \Pi_\Delta{}^T$$

Since M' is a stochastic matrix, the DiffRank model is reduced to a personalized PageRank model for which a unique solution is guaranteed [77, 52]. □

3.3.6 Scalability

While the differential connectivity is computed in a linear time, computing the differential centrality is time consuming because it requires finding the shortest paths between the genes. Using the traditional Dijkstra's algorithm, computing the shortest paths between two nodes requires $O(m + nlog(n))$ where m is the number of links, and n is the number of nodes in the graph and solving all-pairs shortest paths requires $O(nm+n^2logn)$ time and $O(n^2)$ space [49]. However, some recent methods have been proposed to reduce the computational overhead by using approximation methods [49], which can potentially help in efficiently applying *DiffRank* on large-scale networks. In our previous work, we applied the *DiffRank* algorithm in other domains such as the co-authorship networks [96].

3.4 Experiments

Given the i^{th} gene, $k^A(i)$ and $k^B(i)$ are the connectivity of the i^{th} gene in networks A and B, respectively;

1. (Δ **PR**): As a baseline method, we used the difference between the scores computed by the PageRank algorithm [13] in the two networks and is defined as follows:

$$\Delta PR(v) = |PR^A(v) - PR^B(v)| \tag{3.7}$$

Where $PR^K(v)$ is the score for the gene v obtained by applying PageRank on network

K.

2. (**DH**): Differential Hubbing was defined based on the degrees of each gene as follows [60]:

$$DH(v) = K_i^A - K_i^B \tag{3.8}$$

3. (**DC**): Differential Connectivity was defined based on the degrees of each gene as follows [109]:

$$DC(v) = log_{10}(\frac{K_i^A}{K_i^B}) \tag{3.9}$$

4. (**DiffK**) is defined as follows [44]:

$$DiffK(v) = |K^A(v) - K^B(v)| \tag{3.10}$$

where $K^A(v) = \frac{k^A(v)}{max(k^A)}$ and $K^B(v) = \frac{k^B(v)}{max(k^B)}$.

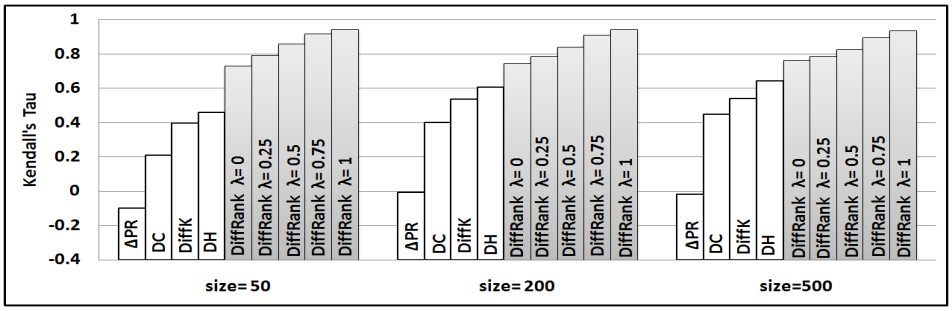


Figure 3.3: Results on simulated networks evaluated based on the local measure (M_L).

3.4.1 Synthetic Differential Scale-free Networks

We developed a simulator to generate synthetic differential scale-free networks. Initially, we started with a small network as a seed; then followed the preferential attachment rule [7] in adding new nodes. This rule assumes the probability of receiving new edges increases with the increase in node degree. To generate two differential networks of size n, we start with the

same seed for each network of size m; then we generate the remaining $n - m$ nodes for each network separately.

Evaluation Measures

Since there is no standard measure for comparing two networks, we developed two evaluation measures, and we used the $Kendall^{'}s\ Tau$ statistic [78] to measure the correlation between the evaluation measures and the ranking algorithms.

Local structure measure (M_L): This measure depends on comparing the edges of each node to find the differential genes. It is a local measure which is defined as follows:

$$M_L(v) = \sum_{u=1}^{N} [w^A(u,v) - w^B(u,v)]^2 \tag{3.11}$$

Global structure measure (M_G): This measure captures the global changes in the gene networks, and it uses the shortest paths in the computation as follows: Let us define $dist(u, v, G^c)$ to be the distance between the nodes u and v in graph G^c computed through the shortest path between them, and let $G_z^{c'}$ be the same as G^c except that all the edges for node z are removed. Then, we define $\Delta_z dist(u, v, G^c) = [dist(u, v, G^c) - dist(u, v, G_z^{c'})]^2$. Finally, M_G is defined as follows:

$$M_G(z) = \sum_{u=1}^{N}\sum_{v=1}^{N} [\Delta dist(u, v, G^A) - \Delta dist(u, v, G^B)]^2 \tag{3.12}$$

M_G measures the importance of each node to all other nodes in the network. It captures the contribution of each gene in the global structure of the network by considering the changes in the shortest paths between each pair of genes.

3.4.2 Results on Simulated Datasets

Figure 3.3 shows the results on the simulated data for different network sizes: 50, 200 and 500 evaluated using M_L. These results are the average of 10 runs. As shown in this figure, it is obvious that as the value of λ increases from 0 to 1, better results are obtained. This is

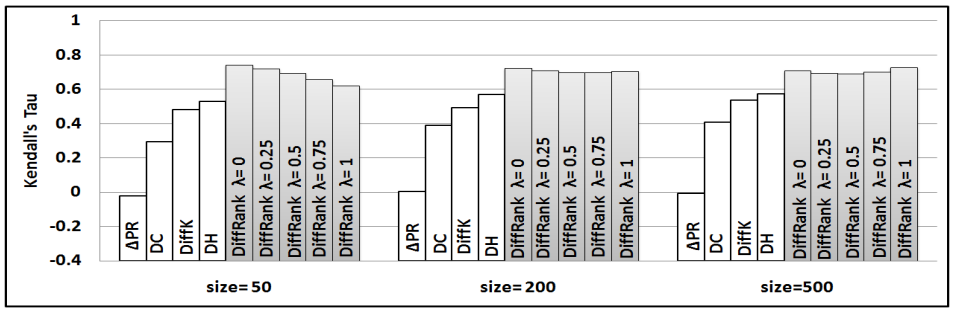


Figure 3.4: Results on simulated networks evaluated based on the global measure (M_G).

because the M_L measure depends only on the connectivity and does not include the centrality component. However, regardless of the value of λ, the *DiffRank* algorithm outperforms the other methods in all of the cases. Figure 3.4 shows the results of the simulated data for different network sizes: 50, 200 and 500 evaluated using M_G. These results are the average of 10 runs. Again, regardless the value of λ, the *DiffRank* algorithm outperforms the other methods in all the cases.

Table 3.1: Description of the four gene expression datasets used in our experiments.

Dataset	**Genes**	**Class A**		**Class B**	
		Description	**Samples**	**Description**	**Samples**
Leukemia [48]	3051	AML	11	ALL	27
Medulloblastoma [91]	2059	Metastatic	10	Non-metastatic	13
Lung cancer [39]	1975	Normal	67	Tumor	102
Gastric cancer [53]	7192	Normal	8	Tumor	22

3.4.3 Experiments on Real-world Datasets

Table 3.1 shows the four real-world datasets used in our experiments. For each dataset, we built a network for each class; then, we ran the proposed method on the two networks.

Constructing the Gene Co-expression Network

Mutual Information (MI) can be used to measure the correlations between different genes, and it outperforms Pearson correlation and other linear measurements because it can capture

Table 3.2: Degree distribution of the networks built for our experiments.

Dataset	**Class**	**Min**	**Mean**	**Max**
Leukemia	AML	5	8.7	96
	ALL	5	8.8	120
Medulloblastoma	Metastatic	5	8.5	66
	Non-metastatic	5	9.0	743
Lung cancer	Normal	5	9.9	878
	Tumor	5	9.9	858
Gastric cancer	Normal	5	9.4	288
	Tumor	5	8.5	248

nonlinear dependencies [128]. Therefore, we used MI to construct the gene networks defined as follows:

$$MI(g_1, g_2) = H(g_1) + H(g_2) - H(g_1, g_2)$$

where H is the entropy, which is calculated as [121]:

$$H(g_1) = -\sum_j P(g_{1_j}) \log P(g_{1_j})$$

$$H(g_1, g_2) = -\sum_i \sum_j P(g_{1_i}, g_{2_j}) \log P(g_{1_i}, g_{2_j})$$

where $P(g_{1_j})$ is the probability that gene g_i takes the value d_{ij}, and $P(g_{1_j}, g_{2_j})$ is the joint probability of the g_1 and g_2 genes.

To find the threshold for the MI values, we followed the rank-based approach that was proposed in [111]. The MI between each gene and all other genes are computed and ranked; then, each gene will be connected to the top d genes that are similar to it. Based on this approach, the minimum degree is d, the mean degree is between d and $2d$ and the maximum degree can be $N - 1$. There are two main advantages of this approach over the other value-based approaches [111]: First, the network will contain only reliable edges. Second, there will be no isolated nodes in the networks. We used $d = 5$, and the resulting networks for each

class are given in Table 3.2. This table shows the minimum, the mean and the maximum of the degrees. However, it is worth mentioning that the proposed algorithm can be applied on any network regardless of the construction method used.

Biological Evaluation

To evaluate the results of proposed algorithm, we used the DAVID functional annotation tool [59] to identify enriched biological GO terms and biological pathways of the top 100 ranked genes in each dataset, and we showed the top five biological terms ranked based on their corrected p-values. In addition, we compared the top 100 ranked genes with the previously published results in the original papers from which we obtained the datasets.

Results

The top 3 differential genes from each dataset are shown in Table 3.3. In this table we present the degrees of each gene in network A, network B and the common edges between the two classes. Table 3.4 shows the top 5 enriched biological terms for each dataset using the DAVID tool [59].

Table 3.3: Top 3 differential genes obtained from the gene expression datasets.

Dataset	**Rank**	**Gene Name**	**Degree in Class A**	**Degree in Class B**	**Common Edges**
Leukemia	1	M26692_s_at	21	92	1
	2	X03934_at	120	5	1
	3	D87459_at	6	96	0
Medulloblastoma	1	196_s_at	5	743	3
	2	2008_s_at	5	709	2
	3	664_at	25	678	6
Lung cancer	1	MTHFR	15	659	11
	2	BAI1	84	492	52
	3	CSF1	530	851	496
Gastric cancer	1	HG1751HT1768_s_at	22	248	0
	2	M10098_5_at	123	224	7
	3	M11722_at	62	181	2

(i) The Leukemia Dataset: The leukemia data contains the expression profiles of 3051 genes in 38 tumor samples. In this dataset, there are 27 acute lymphoblastic leukemia (ALL) samples and 11 acute myeloid leukemia (AML) samples [48]. For this dataset, we applied the version 1 of the proposed *DiffRank* algorithm. In addition to the functional enrichment analysis, we compared our results with the previously published results, and we found some differential genes, such as *M80254_at (CyP3)* and *M27891_at (Cystatin C)*, were reported in [48] among the most highly correlated genes with AML-ALL class distinction.

(ii) The Medulloblastoma Dataset: Medulloblastoma is a common malignant brain tumor of childhood. The medulloblastoma dataset [91] contains gene expression profiles of primary medulloblastomas clinically designated as either metastatic or non-metastatic. For this dataset, we applied the version 1 of the proposed *DiffRank* algorithm and found some statistically significant pathways such as: *Pathways in cancer, Chemokine signaling pathway, MAPK signaling pathway* which have p-values= $1.7E-06,\ 4.0E-04$ and $1.0E-02$, respectively. The mitogen-activated protein kinase **MAPK** signal transduction pathway was reported as an up-regulated pathway in the metastatic tumors that is relevant to the study of the metastatic disease [91]. In addition, some of the top differential genes were reported in [91] among the genes differentiating metastatic from non-metastatic tumors, such as *2042_s_at, 311_s_at* and *1001_at*.

(iii) The Lung Cancer Dataset: This dataset [39] contains the expression profiles of 1975 genes in normal and lung cancer samples. For this dataset, we applied the version 2 of the proposed *DiffRank* algorithm. When compared with the previously published results on the same dataset, we found that some of the top ranked genes, such as $\{$*CLDN14, PAX7, SDCBP, TADA3L, ITGA2B*$\}$, were also reported in the differential patterns discovered by the subspace differential co-expression analysis proposed in [39].

(v) The Gastric Cancer Dataset: The Gastric cancer dataset [53] contains the expression profiles of 7192 genes in normal and Gastric cancer samples. For this dataset, we applied the

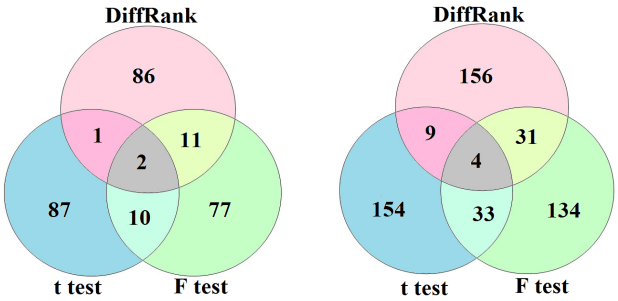
version 2 of the proposed *DiffRank* algorithm and found some of the top ranked genes such as *X51441_s_at* and *Y07755_at* had been reported as highly expressed genes in gastric tumors in [53].

Table 3.4: Top 5 enriched biological terms obtained from the gene expression datasets

Dataset	**Term**	**Fold Enrichment**	**Corrected p-value**
Leukemia	transmembrane protein	4.51	$2.9E-03$
	GO:0005829 cytosol	2.66	$1.1E-02$
	GO:0033273 response to vitamin	15	$1.8E-02$
	GO:0002520 immune system development	5.98	$2.3E-02$
	GO:0048534 lymphoid organ development	6.35	$2.8E-02$
Medulloblastoma	hsa05200:Pathways in cancer	4.83	$1.7E-06$
	kinase	5.47	$4.8E-06$
	ATP	9.75	$1.3E-05$
	domain:Protein kinase	6.64	$1.9E-05$
	nucleotide-binding	3.22	$1.9E-05$
Lung cancer	acetylation	2.73	$2.3E-06$
	Proto-oncogene	10.14	$3.2E-06$
	disease mutation	3.30	$4.1E-06$
	phosphoproteinr	1.71	$4.5E-06$
	nucleus	2.13	$4.9E-06$
Gastric cancer	GO:0005576 extracellular region	2.57	$1.3E-04$
	signal peptide	2.21	$1.3E-03$
	GO:0005615 extracellular space	3.59	$3.1E-03$
	disulfide bond	2.10	$3.5E-03$
	GO:0044459 plasma membrane part	2.0	$4.1E-03$

3.4.4 The Relationships Between DiffRank and Other Approaches

The relationships between the top ranked genes from the *DiffRank* algorithm, DE (represented by the t-test) and DV methods (represented by the F-test) are shown in Figure 3.5. The numbers in this figure are the averages of the rankings from the four datasets. As shown in this figure, most of the genes identified by one approach cannot be identified by the other approaches. This fact explains why we found a few number of genes that were previously published and were top ranked by our algorithm. Furthermore, some of the top ranked genes have not been annotated yet. For example the top ranked gene from the Gastric dataset, $HG1751$-$HT1768\_s\_at$, has no annotations according to the NCBI[1]. As shown in Table 3.3, this gene has 22 edges in the normal network and 248 different edges in the tumor network. From these



(a) Top 100 genes. (b) Top 200 genes.

Figure 3.5: The overlap between the results of the *DiffRank* algorithm, the t-test and the F-test. The numbers are the averages of the four datasets (a) based on the top 100 genes in each method and (b) based on the top 200 genes in each method.

numbers, one can observe that this gene may be involved in important biological processes relevant to the Gastric cancer. Such genes can further be investigated.

3.5 Summary of the *DiffRank* Algorithm

In this chapter, we propose the novel problem of finding the differential hubs in homogenous networks. Given two networks with the same nodes but different edges, the proposed *DiffRank* algorithm can find the differential hubs that are responsible for the differences between the two networks. We make several key observations about how the local and global measures mutually influence the ability to identify the differential nodes, and propose a novel algorithm, called *DiffRank*, for mining the top K differential hubs in the two networks. Comprehensive experimental studies on real-world datasets and synthetically generated datasets showed that our approach outperforms the baselines.

[1] *http://www.ncbi.nlm.nih.gov/*

CHAPTER 4

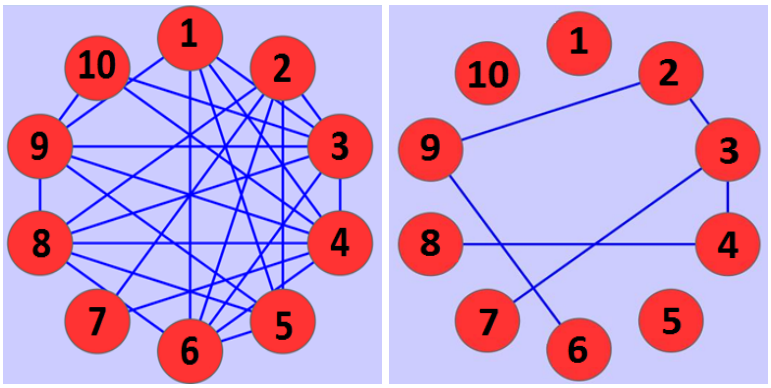
IDENTIFYING DIFFERENTIAL SUBNETWORKS

4.1 Motivation

One of the main goals of using high throughput data such as the DNA microarray is to find disease markers. To achieve this gaol, it is crucial to identify the differences between normal and affected cells [32]. However, it was shown that disease candidate genes are not marked only by the changes in their expression levels, but also by the changes in the gene-gene correlation and the changes in the network structure [90]. Therefore, differential networking is considered as a powerful approach to detect the changes in the network structure and to identify the differentially connected genes among two gene networks. In this approach, a gene co-expression network is constructed for each condition (normal and disease); then, an objective function is optimized to score either single genes (to identify differential hubs) or a group of connected genes (to identify differential subnetworks) based on the differences between the two gene co-expression networks.

The guilt-by-association principle states that genes with similar functions exhibit similar expression patterns (co-expressed) [132, 33]. Therefore, it is crucial to study the relationships between the genes among various biological conditions [2]. Given a gene expression data where the samples belong to one of two biological samples such as normal or cancerous. The author proposes a novel network-based differential subnetwork algorithm to identify differential subnetworks between two networks. A differential subnetwork is a subset of the genes that are strongly connected in one network but not in the other. The proposed algorithm was evaluated on simulated data. As real-world application, we applied and analyzed the proposed differential subnetwork algorithm to the analysis of racial disparity in prostate cancer.

An illustration example of differential subnetworks is shown in Figure 4.1. The shown



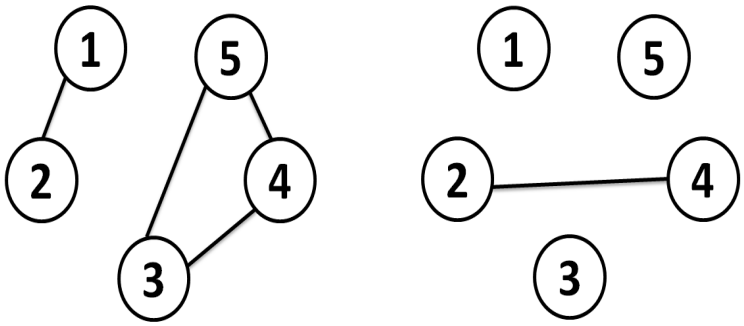
(a) Subnetwork A (b) Subnetwork B

Figure 4.1: A simple illustration of differential subnetworks.

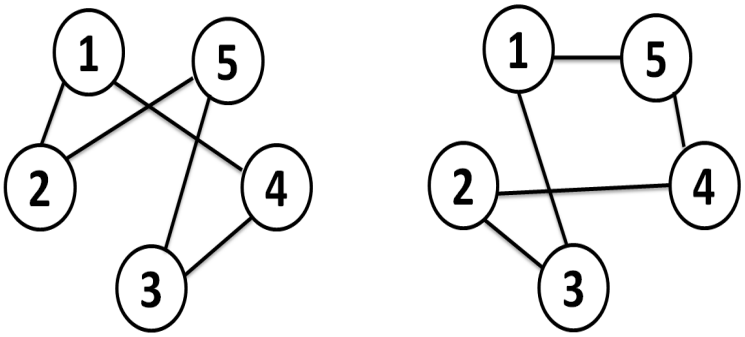
networks were generated using Cytoscape tool [116]. Each subnetwork has the same set of 10 nodes, but the edges between the nodes are different. Basically, the nodes in the first subnetwork are highly inter-connected compared to the second subnetwork. The main characteristics of the differential subnetworks are the following:

- **Differentially connected**. The nodes in a differential subnetwork must be strongly connected in one network but not in the other network.
- **Dense subnetworks**. Differential subnetworks must be dense subnetworks in one and only one network.
- **Overlap**. Differential subnetworks can overlap within the same network, but they should not overlap between the two networks.
- **Hubs and non-hubs**. Differential subnetworks can have both hub and non-hub nodes.

Figure 4.2 shows two examples of non-differential subnetworks. In this Figure, subnetwork 1 is not considered as a differential subnetwork because the nodes in this subnetwork are not strongly connected in any of the two networks although the nodes have more connections in network A than in network B. Subnetwork 2 is not considered as a differential subnetwork



(a) Subnetwork 1 in network A (b) Subnetwork 1 in network B



(c) Subnetwork 2 in network A (d) Subnetwork 2 in network B

Figure 4.2: Examples of two non-differential subnetworks.

because the nodes have the same number of edges in both networks although none of the edges is common between the two networks.

4.2 Related Work

Compared to identifying differential hubs, identifying differential subnetworks is even more challenging since it optimizes for a class-specific group of connected nodes. Hence, rather than considering only the hubs, the goal is to find a group of nodes such that the connectivity and the structure of this group have been significantly changed between the two networks. Most of the existing methods merely perform pairwise comparisons based on: (i) the nodes (jActiveModules [62], DDN [145] and OptDis [31]), (ii) the edges (DifferentialNW [17], Differential clique analysis [132], DiffCoEx [127], postOR [30] and [87]) or (iii) both of the nodes

and the edges (COSINE [90] and PNA [73]). Hence, these methods do not capture the global changes in the network because they focus only on the local comparisons. In addition, most of these methods use some thresholds to define the differential edges or differential nodes. For instance, differential correlation was defined as an edge that was present above 0.875 in one network and the corresponding correlation value in the other network was less than 0.25 [132]. The major problem in such thresholding-based approaches is that it is difficult to accurately determine the optimal values for the thresholds and such methods typically produce different results under different parameter settings. Here, we propose a novel algorithm to identify the differential subnetworks. The proposed algorithm incorporates the differential node scores obtained from the *DiffRank* algorithm described previously.

4.3 The Proposed Differential Subnetwork Algorithm

In this Section, we describe the novel proposed algorithm (*DiffSubNet*) which can be used to identify the most differential subnetworks between two gene subnetworks that represent two phenotypes. In addition, we discuss how to statistically measure the significance of the identified differential subnetworks.

4.3.1 Preliminary and Problem Formulation

Given two gene networks, represented by graphs $G^A(V, E^A)$ and $G^B(V, E^B)$, where V is the set of the N nodes and E^c is the set of edges in G^c, $c \in \{A, B\}$. An edge between two genes u and v, with a weight $w^c(u, v)$ in G^c, determines the strength of the interaction between the genes. Let S denote a subnetwork or a group of connected genes. In addition, we are given the results of the DiffRank algorithm as a vector $\Pi =< \pi_1, \pi_2, ..., \pi_N >$, where π_v denotes the rank of the node v using the DiffRank algorithm, the goal of the *DiffSubNet* algorithm is to find the set of differential subnetworks in class A:

$$< S_1^A, S_2^A, S_3^A ..., S_N^A >$$

and the set of differential subnetworks in class B:

$$< S_1^B, S_2^B, S_3^B ..., S_N^B >$$

where S_m^c denotes the m^{th} differential subnetwork in class c. The proposed algorithm generates N differential subnetworks in each class. For each node n of the N nodes the the network, the proposed algorithm generates the most differential subnetwork containing n as a seed node. The subnetworks are scored and ranked using the following objective function:

$$\Omega(S_m^A) = \sum_{u,v \in S_m} \pi_u \pi_v (w^A(u,v) - w^B(u,v))$$

and for class B:

$$\Omega(S_m^B) = \sum_{u,v \in S_m} \pi_u \pi_v (w^B(u,v) - w^A(u,v))$$

The resulting differential subnetworks in each class are ranked based on the corresponding objective function. The top ranked ones are reported, and the remaining ones are ignored.

4.3.2 The DiffSubNet Algorithm

A differential subnetwork is defined as a subset of highly connected nodes in one network compared to the other network (such as dysregulated pathways). The proposed algorithm (*DiffSubNet*) is described in Algorithm 1.

The proposed *DiffSubNet* algorithm starts with a seed node, then it finds the differential subnetwork that contains that node. It is an iterative algorithm that adds one node to the subnetwork at each iteration. The *DiffSubNet* algorithm produces an initial candidate set to select the node to be added to the subnetwork. This set is composed of all the nodes that are connected to the subnetwork. In the next step, the initial candidate set is filtered by removing the nodes that do not have enough connections with the subnetwork. For this purpose, the proposed algorithm uses a predefined density factor d. If $d = 0.5$, the algorithm excludes the nods that

Algorithm 1 DiffSubNet(G^A,G^B, Π, d)

Input: Data matrix (D)
Network A (G^A)
Network B (G^B)
The DiffRank results ($\Pi =< \pi_1, \pi_2, ..., \pi_N >$)
The subnetwork density factor (d)
Output:: A set of differential subnetworks in class A (S^A)
A set of differential subnetworks in class B (S^B)
Procedure:
for $seed = 1 : N$ **do**
  $/*$ Initialize the current subnetwork in each class to the seed node $*/$
  $S^A_{seed} = \{seed\}$
  $S^B_{seed} = \{seed\}$
  repeat
    $/*$ Define the initial candidate set as the neighbors of the current subnetwork S^A_{seed} $*/$
    $iniCandSet^A = \bigcup\{v : w^A(u,v) > 0, u \in S^A_{seed}, v \notin S^A_{seed}\}$
    $/*$ From the candidate set, exclude the nodes that are less connected to subnetwork S^A_{seed} $*/$
    $candSet^A = \{iniCandSet^A\} - \bigcup\{z : \sum_{u \in S^A_{seed}} |w^A(u,z) > 0| > |S^A_{seed}| * d\}$
    $/*$ Find the best node to be added to S^A_{seed} $*/$
    $y^A = \arg\max_{y_i \in candSet^A} \sum_{u,v \in \{S_m \bigcup y_i\}} \pi_u \pi_v \left(w^A(u,v) - w^B(u,v)\right)$
    $/*$ Add y^A to S^A_{seed} $*/$
    $S^A_{seed} = S^A_{seed} \bigcup y^A$
  until No more nodes can be added to S^A_{seed}
  repeat
    $/*$ Define the initial candidate set as the neighbors of the current subnetwork S^B_{seed} $*/$
    $iniCandSet^B = \bigcup\{v : w^B(u,v) > 0, u \in S^B_{seed}, v \notin S^B_{seed}\}$
    $/*$ From the candidate set, exclude the nodes that are less connected to subnetwork S^B_{seed} $*/$
    $candSet^B = \{iniCandSet^B\} - \bigcup\{z : \sum_{u \in S^B_{seed}} |w^B(u,z) > 0| > |S^B_{seed}| * d\}$
    $/*$ Find the best node to be added to S^B_{seed} $*/$
    $y^B = \arg\max_{y_i \in candSet^B} \sum_{u,v \in \{S_m \bigcup y_i\}} \pi_u \pi_v \left(w^B(u,v) - w^A(u,v)\right)$
    $/*$ Add y^B to S^B_{seed} $*/$
    $S^B_{seed} = S^B_{seed} \bigcup y^B$
  until No more nodes can be added to S^B_{seed}
end for
return $(\{S^A\}, \{S^B\})$

are not connected to at least half of the nodes in the subnetworks. The goal of using this condition is to target dense differential subnetworks. From the filtered candidate set, the *DiffSubNet* algorithm selects the node that maximizes the objective function defined earlier. This process is repeated until no more nodes can be added to the subnetwork. Finally, the subnetworks in each class are ranked based on the objective function. The most differential subnetworks are

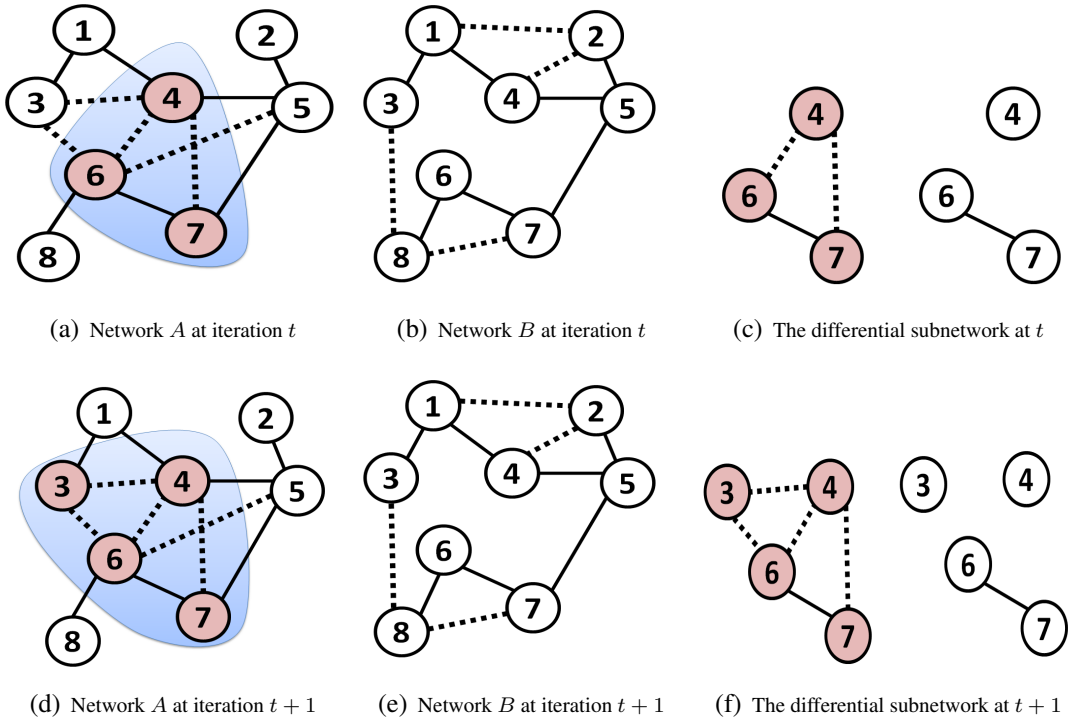
reported and the remaining ones are ignored.

Example. To illustrate how the proposed algorithm works, Figure 4.3 shows an example of two unweighted networks. Each network has 8 nodes. The solid lines between the nodes represent common edges, and the dashed lines represent unique edges. Let us assume that the algorithm at iteration t identified the nodes $\{4, 6, 7\}$ as a differential subnetwork. Figure 4.3(a) shows network A, Figure 4.3(b) shows network B and Figure 4.3(c) shows the differential subnetwork identified from network A at time t. The question is: which node can be added to the current subnetwork at iteration $t + 1$?

The result of the *DiffRank* algorithm $\{8, 2, 3, 7, 4, 1, 6, 5\}$ where node 8 is the top differential hub node in this example. To add a new node to the current differential subnetwork, the *DiffSubNet* algorithm defines the initial candidate set as the neighbors of the current subnetwork (*lines 9-10*). In this example, the initial candidate set is $\{1, 3, 5, 8\}$. Next, the *DiffSubNet* algorithm excludes the nodes that are less connected to the current differential subnetwork (*lines 11-12*). If we assume that the density factor is set to 0.5, then the algorithm excludes each node that is not connected to at least two nodes in the subnetwork. As a result, nodes 1 and 8 will be excluded and the final candidate set is reduced to $\{3, 5\}$. Node 3 has two connections with the current subnetwork, and node 5 has three connections with the current subnetworks. However, since node 5 has two connections that are common in both network A and network B, the *DiffSubNet* algorithm prefers node 3 based on the objective function (*lines 13-14*) because it has a better rank compared to node 5 according to the *DiffRank* algorithm. Hence, at iteration $t + 1$ the differential subnetwork will contain the following nodes: $\{3, 4, 6, 7\}$ as shown in Figure 4.3(d)-(f) .

4.3.3 Evaluation Using Statistical Analysis

For quantitative evaluation, a permutation procedure is performed. Given a differential subnetwork S_n^A of size $|S_n^A|$ genes, its statistical significance can be assessed by randomly permutating the class labels of the samples, and then comparing the differential correlation



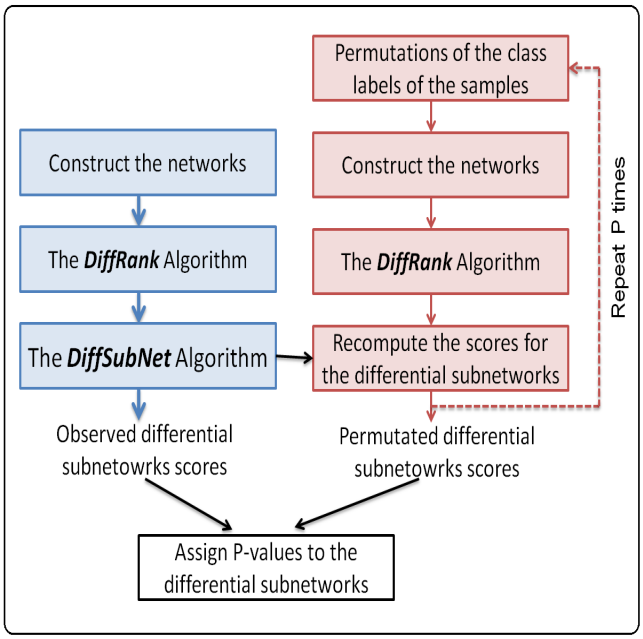
(a) Network A at iteration t (b) Network B at iteration t (c) The differential subnetwork at t

(d) Network A at iteration $t+1$ (e) Network B at iteration $t+1$ (f) The differential subnetwork at $t+1$

Figure 4.3: An illustration example of the *DiffSubNet* algorithm identifies differential subnetworks. The figures in the first row show network A, network B and the differential subnetwork at iteration t, while the figures in the second row show network A, network B and the differential subnetwork at iteration $t+1$.

of the observed and the randomized subnetworks. The differential correlation is measured based on the gene-gene correlations of all the genes in the subnetwork in both classes. The subnetwork S is considered significant if the difference between the gene-gene correlation of the two classes is more than the random subnetworks.

Figure 4.4 illustrates the evaluation process. First, the observed differential scores are computed for all the subnetworks generated by the proposed *DiffSubNet* algorithm. Second, the class labels of the biological samples are permutated P times . In each time, the scores for the subnetworks are recomputed and compared with the observed (original) scores. The p-value for a given differential subnetwork can then be computed as the fraction of times the permutated score was larger than or equal to the observed score. We used a significance level of 0.05 to report the statistically significant differential subnetworks.



(a) Subnetwork B

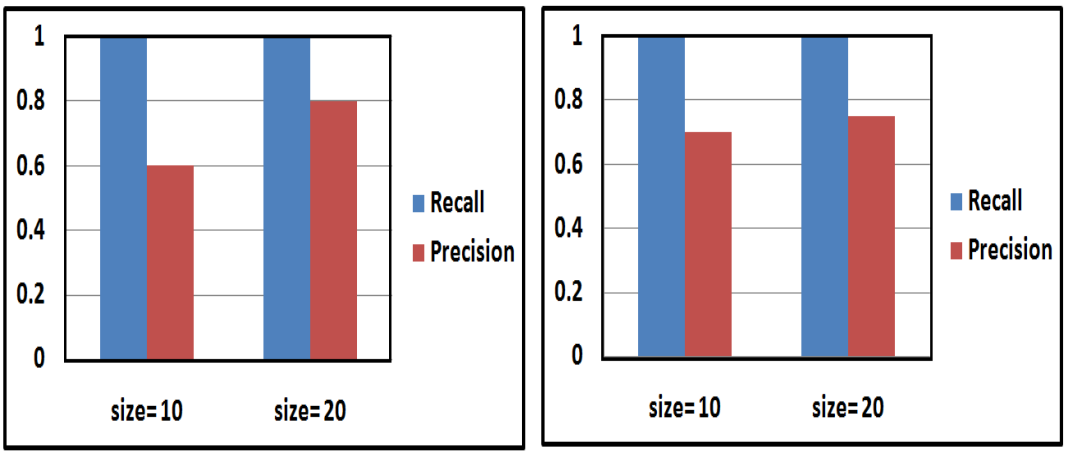
Figure 4.4: Computing the P-values for the differential subnetworks using permutations.

4.4 Experimental Results

In this section, we present the results of the proposed differential subnetwork algorithm on simulated and real dataset. For the simulated dataset, we implanted differential subnetworks in the dataset. As a real-world application, we applied and analyzed the proposed differential subnetwork algorithm to the analysis of racial disparity in prostate cancer.

4.4.1 Constructing the Gene Networks

We used Mutual Information (MI) to measure the correlations between different genes [121] and to construct the gene networks. To find the threshold for the MI values, we followed the rank-based approach that was proposed in [111]. The MI between each gene and all other genes are computed and ranked; then, each gene will be connected to the top 5 genes that are similar to it.



(a) (b)

Figure 4.5: The precision and recall of the simulated datasets. (a) Similar networks. (b) Random networks.

4.4.2 Results on Synthetic Datasets

For the synthetic datasets, a set of subnetworks were implanted in the randomly generated datasets. In the first experiment, we generated a dataset with two classes. The dataset has 1000 genes and 100 samples in each class. The data for each class was randomly generated. We implanted 4 patterns (subnetworks) of size 10 genes and 4 patterns of size 20 genes. These patterns were implanted in one class only (differential). In addition, we implanted some patterns in both classes. These patterns are not differential because they are common in both classes. We constructed a network from each class and run the *DiffSubNet* algorithm, and we evaluated the results using precision and recall which are defined as follows [124]:

$$Precision = \frac{S_{IMP} \cap S_{RES}}{S_{RES}}$$

$$Recall = \frac{S_{IMP} \cap S_{RES}}{S_{IMP}}$$

where S_{IMP} indicates the implanted subnetworks, and S_{RES} indicates the resulting subnetworks. The results are shown in Figure 4.5(a). As shown in this figure, the proposed algorithm

extracted all the implanted differential subnetworks. Therefore the recall value is 1. However, the resulting subnetworks included more nodes than the implanted subnetworks. Since the data was randomly generated, more genes were added to the resulting subnetworks compared to the implanted ones.

In the second experiment, we used exactly the same data for both classes, then we added the differential patterns which were discovered by the proposed algorithm as differential subnetworks. Similar to the first set of experiments, the *DiffSubNet* algorithm successfully identifies the differential subnetworks with some additional nodes as summarized in Figure 4.5(b).

4.4.3 Results on Prostate Cancer

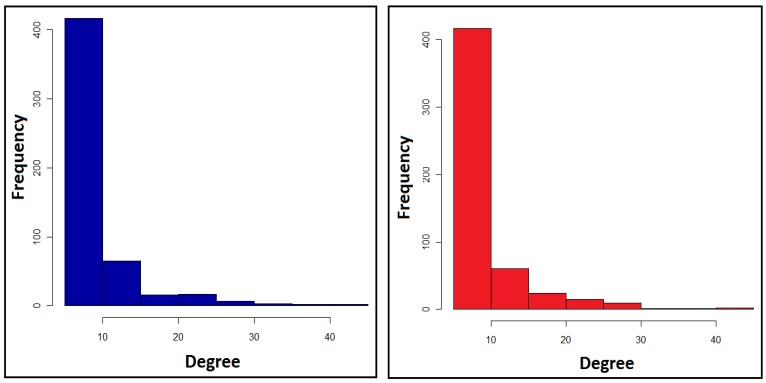
Dataset and Problem Definition

One of the main applications of the DNA microarray data is to compare the biological activities of the genes in two types of cells, such as normal and cancer cells [83]. Comparing the biological roles of genes in two classes of cells is an important problem to identify the genes that are responsible for the phenotypic changes. For instance, African American males (AAM) have a higher risk of developing prostate cancer compared to Caucasian American males (CAM) [107, 133, 110]. There are several hypotheses to explain this difference[66, 110]. One of them is based on the assumption that genetic factors may play a key role in this difference between these two groups. The existing approaches use a simple test, such as the t-test, to identify the differentially expressed genes between AAM and CAM [133, 110]. In this work, we are the first to propose using differential network analysis to identify the genes that are responsible for the differences between Caucasian American and African-American in developing prostate cancer.

The prostate cancer data was generated on a custom Illumina array with 529 genes and 637 samples. This dataset has two classes of conditions. The first class is Caucasian American (369 samples) and the second class is African-American (268 samples).

Results of the *DiffRank* Algorithm

We constructed a gene co-expression network from the Caucasian American expression data and another gene co-expression network from the African-American expression data. Figure 4.6 shows the degree distribution for each network. As shown in this figure, the networks have scale-free structures where most of the nodes in each network have a low number of connections and a few nodes have a high number of connections (hubs).



(a) Caucasian American group (b) African-American group

Figure 4.6: The degree distribution for the prostate cancer networks. (a) Caucasian American group. (b) African-American group.

The next step is running the *DiffRank* algorithm to identify the most differential genes based on their connectivity and centrality which were discussed in the previous chapter. The scores obtained from the *DiffRank* algorithm are used to weigh the nodes when searching for subnetworks. Table 4.1 shows the top 30 genes based on the *DiffRank* algorithm. In addition, it shows the p-values for the t-test and the F-test, respectively. From this table, we emphasize the following observations:

- Some genes are statistically significant based on both tests (Examples include *NFKBIB* and *CAPZB*).
- Some genes are statistically significant based on the t-test but are not statistically significant based on the F-test (Examples include *TCF7L1* and *CD14*).

Table 4.1: The top 30 differential genes in the Prosate cancer dataset.

DiffRank	Gene	P-value t-test	P-value F test
1	*TCF7L1*	0.010	0.570
2	*NFKBIB*	0.002	0.001
3	*CAPZB*	0	0.007
4	*APLP2*	0.234	0.304
5	*CD14*	0	0.704
6	*FOS*	0.033	0.013
7	*ERBB3*	0.038	0.179
8	*NFATC4*	0.035	0.139
9	*AKT1*	0.031	0.566
10	*TGFBR3*	0.359	0
11	*LTC4S*	0.001	0.002
12	*MGP*	0	0.006
13	*ADD2*	0.047	0.185
14	*CCND2*	0.001	0.395
15	*NCDN*	0	0.021
16	*KLK4*	0.290	0.013
17	*CDH1*	0	0.022
18	*PLN*	0.037	0.374
19	*TIMP3*	0.017	0.076
20	*MTHFD2*	0.137	0.171
21	*HPN*	0.369	0.477
22	*ACACA*	0.105	0
23	*KLK2*	0.007	0.019
24	*PCM1*	0.0382	0.187
25	*ERCC2*	0.026	0.461
26	*MYOCD*	0.010	0.010
27	*PLS3*	0.760	0.010
28	*MYLK*	0.001	0.874
29	*TMSB15A*	0.002	0.365
30	*PAICS*	0.019	0

- Some genes are statistically significant based on the F-test but are not statistically significant based on the t-test (Examples include *TGFBR3* and *KLK4*).
- Some genes are not statistically significant based on either the F-test nor the t-test (Examples include *TGFBR3* and *HPN*).

These observations confirmed our hypothesis that differential genes are not marked only by the changes in their expression levels but also by the changes in their connectivity.

Results of the *DiffSubNet* Algorithm

Figure 4.7 shows four examples of differential subnetworks obtained from the Prostatec caner dataset using the proposed *DiffSubNet* algorithm. The first differential subnetwork was obtained from the Caucasian American expression data and it has a p-value of 0. This differential subnetwork is shown in Figure 4.7(a) and in Table 4.2, and the corresponding subnetwork from the African-American expression data is shown in Figure 4.7(b). The second differential subnetwork was also obtained from the Caucasian American expression data and it has a p-value of 0.02. This differential subnetwork is shown in Figure 4.7(c) and in Table 4.3, and the corresponding subnetwork from the African-American expression data is shown in Figure 4.7(d).

Table 4.2: First differential subnetwork in Caucasian American.

Gene	DiffRank	P-value t-test	P-value F test
CAPZB	3	0	0.007
FOS	6	0.033	0.0126
MGP	12	0	0.006
MYLK	28	0.001	0.874
DPYSL3	33	0.008	0
ACTA2	36	0	0
TIMP2	39	0.010	0.028
BLVRA	52	0	0
MAPK8	113	0.059	0.283
EGFR	173	0.228	0.038
HLA-F	218	0.001	0.002

The third differential subnetwork was obtained from the African-American expression data and it has a p-value of 0. This differential subnetwork is shown in Figure 4.7(e) and in Table 4.4, and the corresponding subnetwork from the Caucasian American expression data is shown in Figure 4.7(f). The fourth differential subnetwork was also obtained from the African-

American expression data and it has a p-value of 0.01. This differential subnetwork is shown in Figure 4.7(g) and in Table 4.5, and the corresponding subnetwork from the Caucasian American expression data is shown in Figure 4.7(h).

Table 4.3: Second differential subnetwork in Caucasian American.

Gene	**DiffRank**	**P-value t-test**	**P-value F test**
TCF7L1	1	0.010	0.570
NFATC4	8	0.035	0.139
CCND2	14	0.001	0.395
ACACA	22	0.105	0
TMSB15A	29	0.002	0.365
GATM	73	0.347	0.956
CD40	93	0.937	0.010
GSTP1	115	0.001	0.016
PDGFC	191	0.210	0

These four differential subnetworks were selected based on the objective function (Ω). From these differential subnetworks, we make the following observations:

1. The genes in the differential subnetworks do not necessarily be along to the differential hubs (top ranked by the *DiffRank* algorithm). Most of the differential subnetworks may contain a hub or two hubs, but the remaining genes have lower ranks in the list.

2. The differential subnetworks can overlap. For example, both the differential subnetworks for the African-American group (Table 4.5 and Table 4.5) contain the following genes: *NFKBIB, ERBB3* and *NCDN*. Since the same gene can be involved in more than one biological process or pathway, it is important to develop computational algorithms that allow overlapping patterns or subnetworks.

3. In all of the four differential subnetworks, there are significant differences between the two classes in terms of the connectivity and the structure of the subnetworks. Moreover, in each subnetwork, there is at least one gene that is isolated in the other class. Identifying these isolated genes is very important because each one of them is strongly connected

with a set of genes in one phenotype but is not connected with any gene in that set in the other phenotype.

4. The genes in the differential subnetworks can be statistically significant or insignificant using the standard tests.

Table 4.4: First differential subnetwork in African-American.

Gene	DiffRank	P-value t-test	P-value F test
NFKBIB	2	0.002	0.001
ERBB3	7	0.038	0.179
NCDN	15	0	0.021
ERCC2	25	0.026	0.461
SUFU	50	0.020	0.127
TRAF2	59	0.289	0.202
PLCG1	150	0.469	0.056
PLD2	185	0.010	0.214
TP53	282	0.148	0.045

4.5 Discussion and Summary

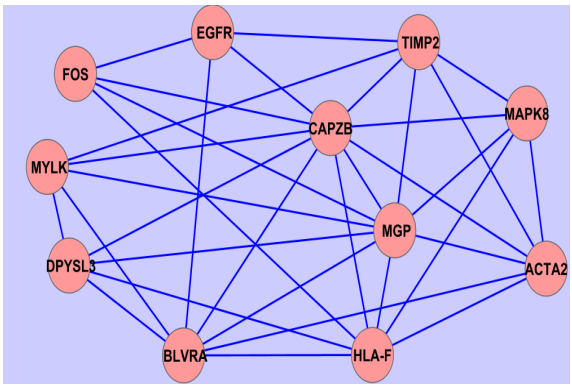
In this chapter, we presented a novel differential network algorithm (*DiffSubNet*) to identify differential subnetworks between two networks that have the same nodes but different set of edges. We demonstrated the effectiveness of the *DiffSubNet* algorithm on simulated data. Moreover, we applied this algorithm to a racial disparity problem, which is a very important problem in bioinformatics. Basically, we are given a dataset that has two classes of biological samples (Caucasian American and African-American). The goal is to study the influence of patient race in the devolvement of Prostate cancer. Although this problem has been tackled by several studies, we are the first to propose solving this problem by using novel differential subnetwork analysis. The differential subnetworks are groups of strongly connected nodes (dense subnetworks) in one network but not in the other network. The resulting differential subnetworks can overlap within the same network, but they should not overlap between the two networks. Furthermore, the genes in the differential subnetworks do not necessarily have

Table 4.5: Second differential subnetwork in African-American.

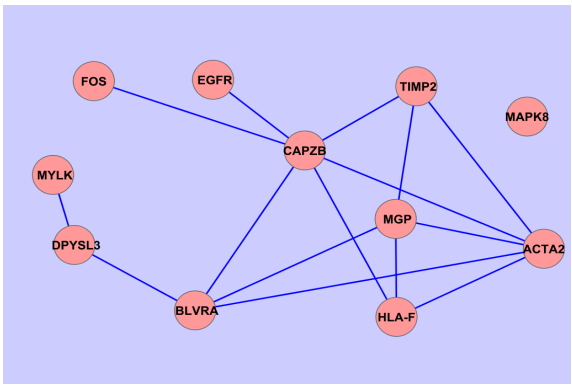
Gene	DiffRank	P-value t-test	P-value F test
NFKBIB	2	0.002	0.001
ERBB3	7	0.038	0.179
AKT1	9	0.031	0.566
NCDN	15	0	0.021
PIK3R2	57	0.075	0.004
PLA2G6	131	0.252	0.052
SEPT5	212	0.783	0.013
GRN	320	0.055	0.005

to be among the differential hubs (top ranked by the *DiffRank* algorithm).

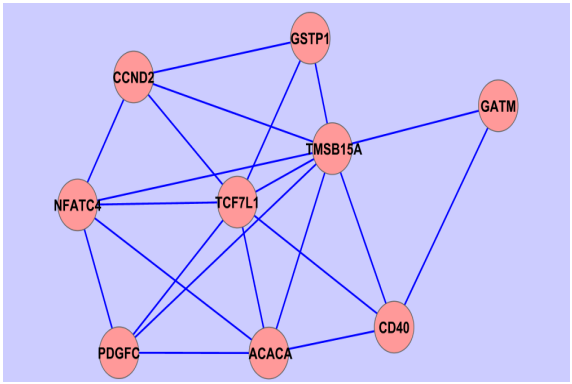
The genes in the differential subnetworks can be statistically significant or insignificant using the standard t-test or the F-test. These tests capture the changes in the expression levels of single genes while the proposed differential subnetwork algorithm captures the changes in the gene-gene correlations and the changes in the connectivity and the structure of the networks. Comprehensive studies should consider all of these changes rather than using a single method of differential analysis.



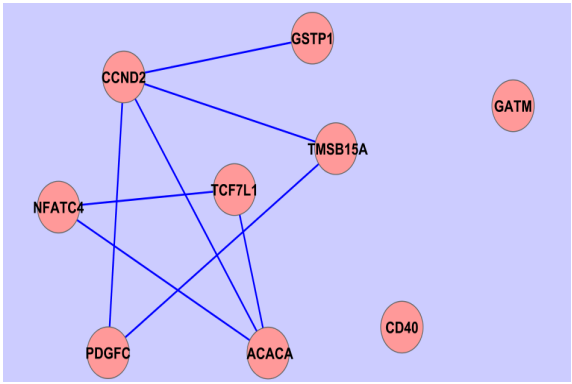
(a) Subnetwork 1 in Caucasian American



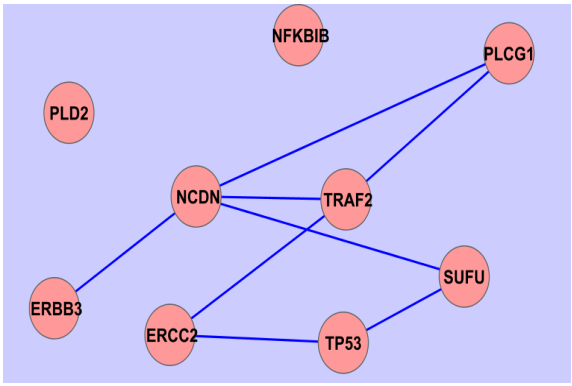
(b) Subnetwork 1 in African-American



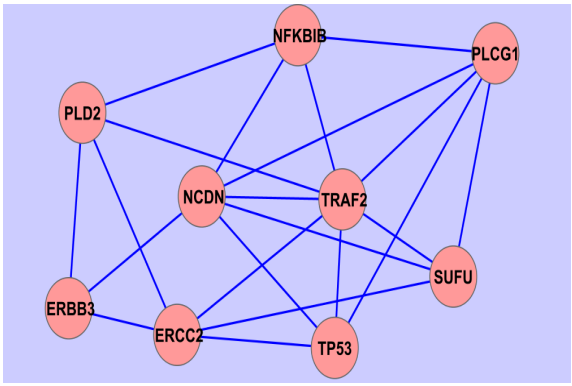
(c) Subnetwork 2 in Caucasian American



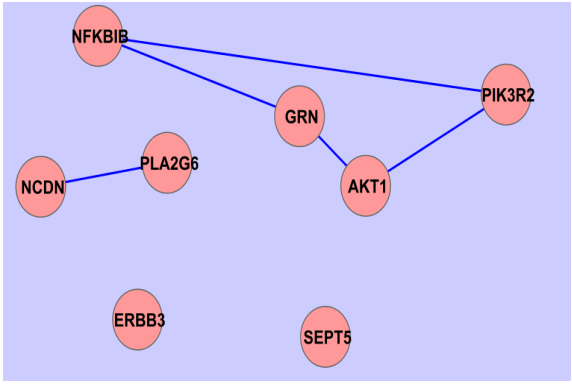
(d) Subnetwork 2 in African-American



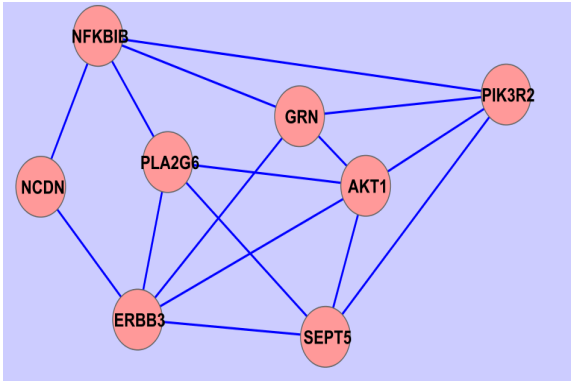
(e) Subnetwork 3 in Caucasian American



(f) Subnetwork 3 in African-American



(g) Subnetwork 4 in Caucasian American



(h) Subnetwork 4 in African-American

Figure 4.7: The top two differential subnetworks in each phenotype.

CHAPTER 5

RANKING-BASED CO-CLUSTERING OF MICROARRAY DATA

5.1 Introduction

The goal of co-clustering (or biclustering) is to simultaneously cluster both rows and columns in a given matrix [22]. Motivated by several applications in text mining, recommendation systems and bioinformatics, different methods have been developed to discover local patterns that cannot be identified by traditional clustering algorithms. In spite of vast research in this research, existing co-clustering algorithms have some critical limitations in terms of identifying co-clusters with different types of correlations in the data and their ability to capture overlapping co-clusters in the data. In this chapter, we present a new deterministic co-clustering algorithm that can be used to efficiently extract significant co-clusters. Our algorithm uses a novel ranking-based objective function that is optimized to simultaneously find large co-clusters with minimum residual errors. It allows positively and negatively correlated objects to be members of the same co-clusters and can extract overlapping co-clusters. In addition, the co-clusters can be arbitrarily positioned in the data matrix.

5.1.1 Motivation

Clustering is an important tool in unsupervised learning that is used to group similar data points [95]. Partitioning data points into clusters is a challenging problem in several data analysis including text mining and bioinformatics. Traditional one-dimensional clustering algorithms, such as $k-$means and hierarchical clustering, assign every data point to a cluster based on a similarity measure computed across all the features. In some applications, traditional clustering algorithms cannot capture the structural patterns in the data [4]. Since these algorithms assume that correlated rows (columns) share similar patterns across all the columns (rows),

they fail to discover local subspace patterns that exist in subsets of rows (or columns) [34] .

Given a data matrix with two entities (objects, features), such as $(words, documents)$ in text mining, $(users, movies)$ in recommendation systems and $(genes, samples)$ in bioinformatics, a subset of rows may be inter-related under a subset of columns forming blocks of substructures (co-clusters). For example, a set of genes may be co-expressed under a subset of samples and applying traditional clustering techniques cannot capture such blocks [4]. Co-clustering has emerged as a powerful tool to simultaneously cluster both dimensions of a data matrix by utilizing the relationship between the two entities [113]. Co-clustering helps in discovering local patterns that cannot be identified by the traditional one-way clustering algorithms.

Compared to traditional one-dimensional clustering, co-clustering is considered more informative and more scalable [6] because it simultaneously measures the degree of coherence in the samples across various attributes of a given data matrix. [57]. Moreover, considering co-clusters rather than the entire feature space reduces the noise that is inherent in the data [46]. Co-clustering has been used in several applications such as clustering microarray data [92], identifying protein interactions [80], collaborative filtering [45], text mining [16], matrix approximation [113]. In this work, we focus on applying co-clustering in biological applications such as gene expression data analysis to identify local patterns.

5.1.2 Characteristics of Co-clusters

There are several important characteristic that should be considered while searching for co-clusters in gene expression data. A subset of genes can be correlated only in a small subset of samples due to the heterogeneity of the samples. Moreover, a gene can be involved in more than one biological pathway; therefore, there is a need for a co-clustering algorithm that allows overlapping between the co-clusters [34], i.e., the same gene can be a member of more than one co-cluster. In addition, since genes can be positively or negatively correlated [67], it is important to allow both types of correlation in the same co-cluster. Furthermore, the

co-clusters can be arbitrarily positioned in the gene expression data. Existing algorithms do not incorporate all of these important characteristics. The proposed algorithm supports the discovery of large and possibly overlapping co-clusters that contain positively and negatively correlated genes. Here, we describe the important characteristics of the co-clusters in the gene expression domain. However, many of these characteristics are applicable to several other domains as well.

1. **Arbitrarily positioned co-clusters.** Due to the heterogeneity of the samples, a subset of genes can be correlated across any subset of the samples. Hence, the co-clusters can be arbitrarily positioned in the matrix [95].

2. **Overlapping.** Discovering overlapping patterns is a challenging task in data mining [89]. For example, a gene can be involved in more than one biological process. Therefore, that gene can belong to more than one co-cluster. One of the main advantages of our algorithm is that it allows overlapping between co-clusters, which helps in understanding the different roles played by a particular gene in a living cell. [95, 34].

3. **Positive and negative correlations.** There are different types of correlations between the genes in any cell. Examples of such relationships are positive and negative correlations [143]. Figure 5.1 shows an example these correlations. In a positive correlation, genes show similar patterns while in a negative correlation, genes show opposite patterns. Since it is possible that genes with both types of correlations exist in the same biological pathway [67], there is a need for a computational model that captures both types of correlations simultaneously [143]. However many of the existing co-clustering algorithms capture positive correlations only. In this chapter, we introduce a novel algorithm that can be used systemically to capture positive and negative correlations simultaneously.

4. **Noisy data.** The expression data contains a huge amount of noise [68]. Hence, the co-clustering algorithms should be robust against noise.

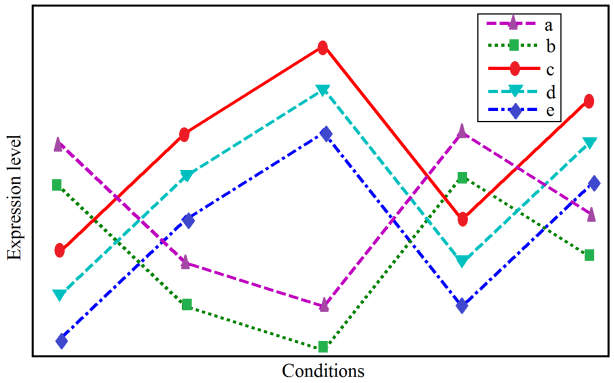


Figure 5.1: Different types of relationships between the genes in one co-cluster. The genes $\{a, b\}$ are positively correlated with each other, and the genes $\{c, d, e\}$ are positively correlated with each other. However, the genes $\{a, b\}$ are negatively correlated with the genes $\{c, d, e\}$.

Measuring the Coherence of Co-clusters

The coherence is a measure of how similar a set of gene expression profiles are. Cheng and Church proposed the mean-squared residue (MSR) score as a measure of the coherence for a given co-cluster [22]

Definition 1. *(***Mean-Squared Residue***). The mean-squared residue of a co-cluster X of $|I|$ rows and $|J|$ columns is measured as:*

$$MSR(X) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (X_{ij} - X_{Ij} - X_{iJ} + X_{IJ})^2$$

where X_{ij} is the value in row i and column j in co-cluster X, $X_{iJ} = \frac{\sum_{j \in J} X_{ij}}{|J|}$ is the row mean, $X_{Ij} = \frac{\sum_{i \in I} X_{ij}}{|I|}$ is the column mean and $X_{IJ} = \frac{\sum_{i,j} X_{ij}}{|I||J|}$ is the overall mean of X.

5.1.3 Our Contributions

In this chapter, we present a novel co-clustering algorithm to efficiently find arbitrarily positioned co-clusters in the data matrix. Our contributions can be summarized as follows:

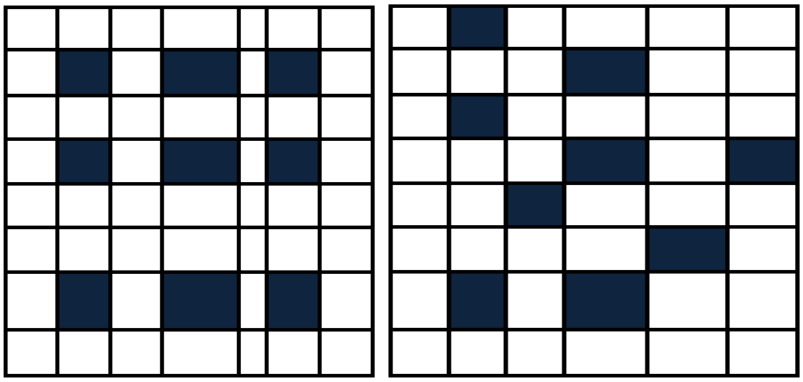
- Propose a novel co-clustering algorithm, **R**anking-based **A**rbitrarily **P**ositioned **O**verlapping **C**o-Clustering (**RAPOCC**), to efficiently extract significant co-clusters.

- Propose a novel ranking-based objective function to find arbitrarily positioned co-clusters.
- Extract large and overlapping co-clusters containing both positively and negatively correlated rows.

5.2 Limitations of Existing Co-clustering Algorithms

In this section, we describe some of the popular co-clustering algorithms. Cheng and Church (CC) [22] proposed the first co-clustering algorithm that produces one co-cluster at a time. The obtained co-cluster is replaced with random numbers, which typically reduces the quality of the co-clusters. The Order-Preserving Submatrices (OPSM) algorithm [11] finds one co-cluster at a time in which the expression levels of all genes induce the same linear ordering of the experiments. A co-cluster is considered order-preserving if there is a permutation of its columns under which the sequence of values in every row is strictly increasing. This algorithm does not capture the negatively correlated genes. The Iterative Signature Algorithm (ISA) [64] defines a co-cluster as a co-regulated set of genes under a set of experimental conditions. It starts from a set of randomly selected rows that are iteratively refined until they are mutually consistent. The Robust Overlapping Co-clustering (ROCC) algorithm [34] finds $\kappa \times \ell$ co-clusters using the Bregman co-clustering algorithm [6]. This algorithm does not handle the negative correlations. Our proposed co-clustering algorithm overcomes all of the above limitations by (i) capturing arbitrarily positioned co-clusters, (ii) handling overlapping and positive and negative correlations and (iii) being robust against noise.

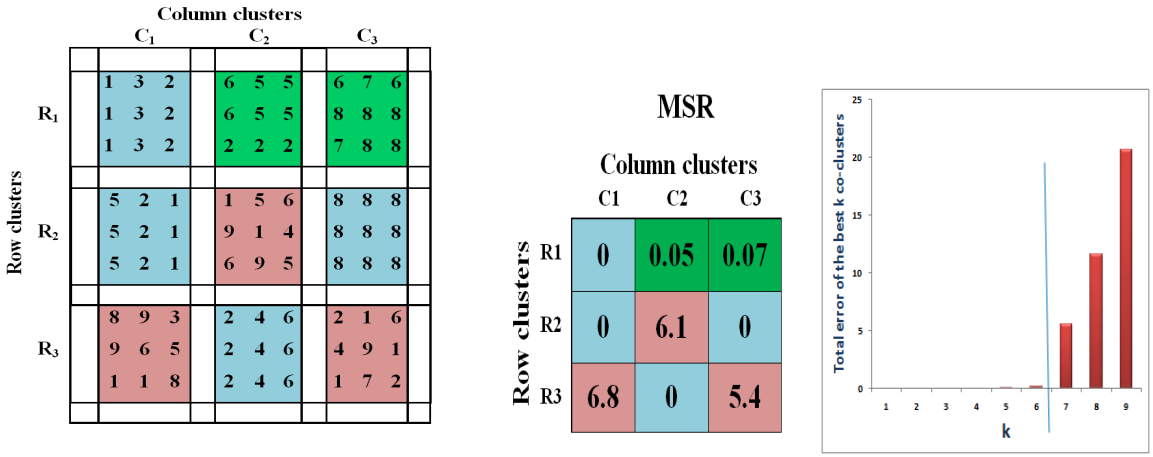
Recently, the (κ, ℓ) co-clustering model has been proposed to simultaneously find $\kappa\ell$ co-clusters [4, 34]. This model was shown to perform well in various applications [4, 34]. However, the main limitation of this model is that it assumes a grid structure comprised of $\kappa \times \ell$ co-clusters as shown in Figure 5.2(a). The assumption here is that the rows in each row cluster should be correlated under each of the ℓ column clusters. Such an assumption may not hold when a subset of rows is correlated only in a limited subset of columns (or vice versa). To



(a) Grid structure (b) Arbitrarily positioned co-clusters

Figure 5.2: Types of co-cluster structures.

overcome this limitation, we propose a novel co-clustering algorithm that is able to identify arbitrarily positioned co-clusters as shown in Figure 5.2(b). We will now discuss two synthetic examples that motivate the need for a new co-clustering algorithm.



(a) (b) (c)

Figure 5.3: Motivating example 1: (a) Nine co-clusters arranged in a 3×3 grid structure. (b) The error of each co-cluster measured by MSR. (c) The accumulated sum of the error of the best K co-clusters is shown in the Y-axis. The value of K is shown on the X-axis (the cut-off is based on elbow point criterion).

Motivating Example 1:

In Figure 5.3(a), an example of 9 co-clusters arranged in a 3×3 grid structure is shown. The corresponding error for each co-cluster is shown in Figure 5.3(b). The error is measured by the mean-squared residue (MSR) score given by Definition (1). Let us consider the co-cluster present in the intersection of the third row cluster and the second column cluster. This co-cluster has an error of 0, which means that this is a perfect co-cluster. However, since the other co-clusters in the same row cluster have high error values, this co-cluster will not be extracted by the existing algorithms. Our objective function depends only on the score of the top-ranked co-clusters. Hence, in this example if 70% of the co-clusters are included in the objective function (as represented by the vertical line shown in Figure 5.3(c)), then the proposed algorithm will be able to identify the six best co-clusters regardless of the score of the three remaining co-clusters. The co-clusters found by our algorithm are the highly ranked ones which are unknown in advance, arbitrarily positioned, and can be changed during the iterative re-assignment step.

Motivating Example 2:

Figure 5.4 shows two co-clusters of size 4×4. The MSR of the first co-cluster is 0.098, and the MSR of the second co-cluster is 2.723, which means that the first co-cluster is more homogenous than the second one. Given a new row, as shown in Figure 5.4, the question is: can we add it to the current co-clusters or not? If this row is to be added; then only the error of the first co-cluster will be reduced. Specifically, the MSR of the first co-cluster will be reduced to 0.085, but the error of the second co-cluster will be increased to 4.47. That is, the average MSR of the two co-clusters before adding the new row is 1.41 while the average MSR of the two co-clusters after adding the new row is 2.273. Therefore, the row will not be added to the current co-clusters because of the high error of the second co-cluster, which will be pruned eventually. In this work, we propose a new objective function that considers the score of the

top-ranked co-clusters when the rows (or columns) are to be added/removed. Therefore, when our algorithm is applied to this example, this row will be added because it improves the score of the co-cluster that already has the maximum score. We will show that by using the new objective function, it is possible to obtain improved results by focusing on the discovery of high quality co-clusters.

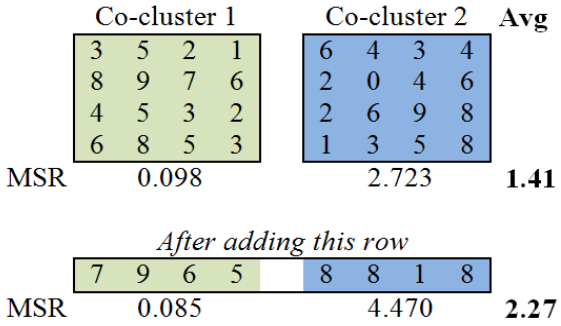


Figure 5.4: Motivating example 2: Two co-clusters are shown with their corresponding MSR. The problem here is to decide whether to add the new row to the current solution or not.

The intuition behind considering only the top-ranked co-clusters in the computation of the objective function is that the co-clusters with high error values will be pruned eventually, and we are interested in finding the co-clusters with the minimum error values regardless of the other co-clusters. The existing co-clustering algorithms optimize for the co-clusters whose sum of errors is minimized, while our algorithm optimizes for the best co-clusters, which could be missed by other algorithms as a result of the effects including the co-clusters with high error values. The set of co-clusters that are found by our algorithm are the highly ranked ones which are unknown in advance, arbitrarily positioned and can be changed during the optimization process.

5.3 The Proposed RAPOCC Algorithm

In this Section, we describe the *RAPOCC* algorithm. This algorithm is proposed to efficiently extract the most coherent and large co-clusters that are arbitrarily positioned in the data

matrix. These co-clusters can overlap and have positively and negatively correlated rows.

5.3.1 Preliminaries

In this section, we introduce the coherence measure that can be used to measure the quality of the co-clusters, and we formulate the problem of co-clustering. The notations used in this chapter are described in Table 5.1.

Table 5.1: Notations used in this chapter.

Notation	Description
D	input data matrix of M rows and N columns
κ	number of row clusters
ℓ	number of column clusters
ρ	mapping of row clusters
γ	mapping of column clusters
K	number of optimized co-clusters
X	Co-cluster of $\|I\|$ rows and $\|J\|$ columns
I	set of rows in co-cluster X
J	set of columns in co-cluster X
x_j	the j^{th} column in row x
$\|.\|$	the cardinality function

5.3.2 Definitions and Problem Formulation

Coherence is a measure of how similar a set of gene expression profiles are. Cheng and Church proposed the mean-squared residue (MSR) score as a measure of coherence [22]. Since the the overall shapes of gene expression profiles are of greater interest than the individual magnitudes of each feature [68], we normalize the expression values of each gene to be between 0 and 1. As a result, the value of the objective function will also be bounded between 0 and 1.

Definition 2. *(***Coherence measure H***). The coherence of a co-cluster X of $|I|$ rows and $|J|$ columns is measured as*

$$H(X) = 1 - \frac{1}{|I||J|} \sum_{i \in I, j \in J} (X_{ij} - X_{Ij} - X_{iJ} + X_{IJ})^2$$

where X_{ij} is the value in row i and column j in co-cluster X, $X_{iJ} = \frac{\sum_{j \in J} X_{ij}}{|J|}$ is the row mean,

$X_{Ij} = \frac{\sum_{i \in I} X_{ij}}{|I|}$ *is the column mean and* $X_{IJ} = \frac{\sum_{i,j} X_{ij}}{|I||J|}$ *is the overall mean of* X.

Using Definition 2, a perfect co-cluster will have a score $= 1$. Given two rows (x and y) and J columns, the coherence measure can be re-written as follows:

$$\begin{aligned} h(x, y, J) &= 1 - \frac{1}{2|J|} \sum_{j \in J} \left(x_j - \bar{x} - \frac{x_j + y_j}{2} + \frac{\bar{x} + \bar{y}}{2} \right)^2 \\ &\quad - \frac{1}{2|J|} \sum_{j \in J} \left(y_j - \bar{y} - \frac{x_j + y_j}{2} + \frac{\bar{x} + \bar{y}}{2} \right)^2 \\ &= 1 - \frac{1}{|J|} \sum_{j \in J} \left(\frac{(x_j - \bar{x}) - (y_j - \bar{y})}{2} \right)^2 \end{aligned} \tag{5.1}$$

where $\bar{x}$ ($\bar{y}$) represents the mean of the values for the row x (y). An optimal co-cluster has a value of $H(X) = 1$, which results from the case where $(x_j - \bar{x}) = (y_j - \bar{y}), \forall j \in J$. This type of correlation is positive ($h_+(x, y, J)$). In the negative correlation, the rows have opposite patterns (i.e. the two negatively correlated rows will get a perfect score when $(x_j - \bar{x}) = -(y_j - \bar{y})\ \forall j \in J$). The positive and negative correlations are defined in Definition 3.

Definition 3. *(***Positive and negative correlations***). Given two rows (x and y) and J columns, the positive correlation between them is defined as*

$$h_+(x, y, J) = 1 - \frac{1}{|J|} \sum_{j \in J} \left(\frac{(x_j - \bar{x}) - (y_j - \bar{y})}{2} \right)^2$$

and the negative correlation is defined as

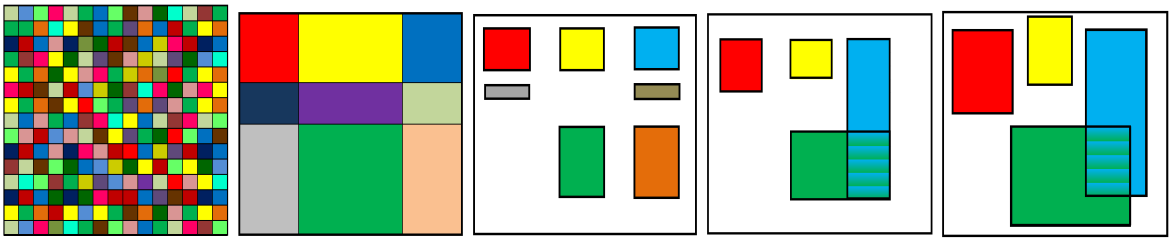
$$h_-(x, y, J) = 1 - \frac{1}{|J|} \sum_{j \in J} \left(\frac{(x_j - \bar{x}) + (y_j - \bar{y})}{2} \right)^2$$

Definition 4. *(***Pairs-based Coherence HP***). Given a co-cluster X of $|I|$ rows and $|J|$ columns, the coherence of this co-cluster is measured based on all the pairs in X:*

$$HP(X) = \frac{|2|}{|I|(|I| - 1)} \sum_{x,y \in X} (h_\circ(x, y, J))$$

where $\circ \in \{-, +\}$.

The type of correlations (either positive or negative) between any two rows, referred to as $\circ$ in Definition 4, is maintained for each pair of rows in each co-cluster in the proposed algorithms.



(a) Input data matrix (b) **Step 1**: Initialization (c) **Step 2**: Core Co-clustering (d) **Step 3**: Merging (e) **Step 4**: Refinement

Figure 5.5: The main steps of the proposed *RAPOCC* algorithm

Now, we will formally define the problem of co-clustering.

Definition 5. *(***Co-clustering***). Let* $D \in \mathbb{R}^{M \times N}$ *denote a data matrix; the goal of co-clustering is to find a row mapping* (ρ) *that maps the rows to the* κ *row clusters and a column mapping* (γ) *that maps the columns to the* ℓ *column clusters*

$$\rho : \{1, 2, ..., M\} \longrightarrow \{1, 2, ..., \kappa\}$$

$$\gamma : \{1, 2, ..., N\} \longrightarrow \{1, 2, ..., \ell\}$$

such that the coherence of the top-K co-clusters is maximized.

$$\underset{X_1, X_2, ..., X_K}{\arg\max} \sum_{i=1}^{K} HP(X_i)$$

The problem of finding the co-clusters is an NP-hard problem [22]. We propose a novel co-clustering algorithm to efficiently find arbitrarily positioned co-clusters from a given data matrix.

5.3.3 Ranking-based Objective Function

In the proposed iterative algorithm, the score of each of the $\kappa\ell$ co-clusters is computed at each iteration, and the overall value of the objective function is computed based on the coherence score of the top-K scores where K is the number of optimized co-clusters ($1 \leq K \leq \kappa * \ell$).

$$\underset{X_1,X_2,...,X_K}{\arg\max} \sum_{i=1}^{K} HP(X_i)$$

The set of the top-K co-clusters can be any subset of the $\kappa * \ell$ co-clusters. *During each iteration, the objective function will be computed for each possible change in the row/column mapping keep the function monotonically increasing*. The advantage of using this objective function is that it allows the discovery of arbitrarily positioned co-clusters.

5.3.4 The RAPOCC Algorithm

The main steps of the *RAPOCC* algorithm are shown in Figure 5.5. The algorithm starts with a two-dimensional matrix (*objects* $\times$ *features*) as an input. In the first step (see Figure 5.5(b)) a divisive approach is used for initialization. Basically, it starts with all the rows and columns in one co-cluster; then the algorithm splits the co-cluster with the largest error. This iterative procedure continues until κ row clusters and ℓ column clusters are obtained. The core co-clustering step (see Figure 5.5(c)) finds the optimal row and column clusterings (ρ, γ). In the third step, Figure 5.5(d), similar co-clusters are merged using a hierarchical agglomerative approach. In the fourth step (see Figure 5.5(e)) more rows and columns are added to each co-cluster individually. Finally, a pruning step is used to prune the co-clusters with low coherence scores. These steps are described in Algorithm 2. In this algorithm, $H(u, v)$ and $HP(u, v)$ indicate the coherence of the co-cluster formed by the row cluster u and column cluster v. The inputs to this algorithm include the data matrix $D \in \mathbb{R}^{M \times N}$, the number of row clusters κ and the number of column clusters ℓ. These are common parameters in the co-clustering methods [34], and they can be set based on the size of the data matrix. K determines the number of the optimized co-clusters and can be set to any value between 1 and $\kappa \times \ell$. The

parameters κ, ℓ and K can be set to large values because the *RAPOCC* algorithm will only report the most coherent co-clusters, and the remaining ones will be pruned in the last step.

Algorithm 2 RAPOCC(D,κ, ℓ,K)

Input: Data matrix (D)
No. of row clusters (κ)
No. of column clusters (ℓ)
No. of optimized co-clusters (K)
Output:: A set of K co-clusters ($\{X\}$)
Procedure:
$\mathbf{Step\ 1}$: initialization
$i \leftarrow 1,\ j \leftarrow 1$
$\rho(g) \leftarrow i, \forall [g]_1^m$
$\gamma(c) \leftarrow j, \forall [c]_1^n$
while $i < \kappa$ **or** $j < \ell$ **do**
  if $i < \kappa$ **then**
    $i \leftarrow i + 1$
    $\alpha \leftarrow \arg\min_\alpha \sum_{j=1}^{\ell} H^{'}(u,v)\ : \rho(u) = \alpha, \gamma(v) = l$
    Partition α using bisecting clustering algorithm
  end if
  if $j < \ell$ **then**
    $j \leftarrow j + 1$
    $\beta \leftarrow \arg\min_\beta \sum_{i=1}^{\kappa} H^{'}(u,v)\ : \rho(u) = i, \gamma(v) = \beta$
    Partition β using bisecting clustering algorithm
  end if
end while
$\mathbf{Step}\ 2$: core co_clustering
repeat
  $/*$ Row clustering $*/$
  for $a = 1 : M$ **do**
    $\rho(a) = \arg\max_{u \in \{-\kappa,\ldots,-1,0,1,\ldots,\kappa\}} HP(\rho(a) = u, \gamma)$
  end for
  $/*$ Column clustering $*/$
  for $b = 1 : N$ **do**
    $\gamma(b) = \arg\max_{b \in \{0,1,\ldots,\ell\}} HP(\rho, \gamma(b) = v)$
  end for
until convergence
$\mathbf{Step\ 3}$: Merging similar co_clusters
$\mathbf{Step\ 4}$: Refinement
$\mathbf{Step\ 5}$: Pruning

Step 1: Initialization. Inspired by the bisecting K-means clustering technique [120], we use a deterministic algorithm for the initialization. Each row is mapped to one of the κ clusters,

and each column is mapped to one of the ℓ clusters, resulting in a checkerboard structure $\kappa \times \ell$ as shown in Figure 5.5(b). The initialization algorithm is a divisive algorithm that starts with the complete data assigned to one cluster as described in Algorithm 2 (*lines 5-7*); then, the following steps are repeated until the desired number of row clusters is obtained. (1) Find the row cluster with the lowest coherence score (α_{min}). (2) Find the two rows in α_{min} with the lowest correlation (r_1, r_2). (3) Create two new row clusters α_1 and α_2. Add r_1 to α_1 and r_2 to α_2. (4) Add each of the remaining rows in α_{min} to α_1 (α_2) if it is more correlated to r_1 (r_2). The column clusters are initialized in the same manner. The algorithm alternates between clustering the rows and the columns as described in Algorithm 2 (*lines 8-19*).

Step 2: Core Co-clustering (ρ, γ). This step finds the optimal row and column clusterings (ρ, γ) as shown in Figure 5.5(c). To update ρ, each row (r_i) is considered for one of the following three actions as described in Algorithm 2 (*lines 20-30*):

- Exclude r_i from any row cluster by setting ρ to 0.
- Find the best row cluster to include r_i as a **positively correlated** row $\{1, 2, .., \kappa\}$.
- Find the best row cluster to include r_i as a **negatively correlated row** $\{-\kappa, ..., -2, -1\}$.

The objective function is computed for each possible action, and the action to be carried out is the one corresponding to the maximum value of the three objective function values. Within each co-cluster, there is a sign vector that determines the type of correlation (positive or negative) of each row. Therefore, a row can be positively correlated in some of the co-clusters and negatively correlated in other co-clusters. The column mapping (γ) is calculated in a similar manner, but there is no consideration for negatively correlated columns. Following this strategy, the value of the objective function is monotonically increasing, and the convergence is guaranteed as shown in Theorem 1. After convergence, the result will be a non-overlapping set of co-clusters.

Theorem 1. *The Algorithm RAPOCC (Algorithm 2) converges to a solution that is a local optimum.*

Proof. From Definition 4, the coherence measure HP is bounded between 0 and 1. Hence, the objective function given in Definition 5 is also bounded. Algorithm 2 iteratively performs a set of update operations for the row clustering and the column clustering . In each iteration, it monotonically increases the objective function. Since this objective function is bounded for the top-K co-clusters, the algorithm is guaranteed to converge to a locally optimal solution. □

Step 3: Merging the Co-clusters. The top-K co-clusters with the maximum coherence are retained from the previous step. In this step, similar co-clusters are merged as shown in Figure 5.5(d) using an agglomerative clustering approach. The two most similar co-clusters are merged in each iteration. The goal of this step is two-fold: (i) it allows the discovery of **large** co-clusters, and (ii) it allows for **overlapping** co-clusters.

Step 4: Refinement. In this step, the algorithm adds more rows and columns to each co-cluster individually to obtain **larger** co-clusters and also allows for **overlapping** co-clusters as shown in Figure 5.5(e). Hence, the same row/column can be added to several co-clusters.

Step 5: Pruning. In this step, we prune the co-clusters with the lowest coherence scores. To determine which co-clusters to prune, (i) sort the co-clusters based on their coherence (measured by HP), (ii) compute the difference between the consecutive scores and (iii) report the set of co-clusters just before the largest difference, and prune the remaining co-clusters. The time complexity of the *RAPOCC* algorithm is $O\left(\kappa.\ell.max(MN^2, NM^2)\right)$.

5.4 The Experimental Results

To demonstrate the effectiveness of the proposed algorithm, several experiments were conducted using both synthetic and real-world gene expression datasets.

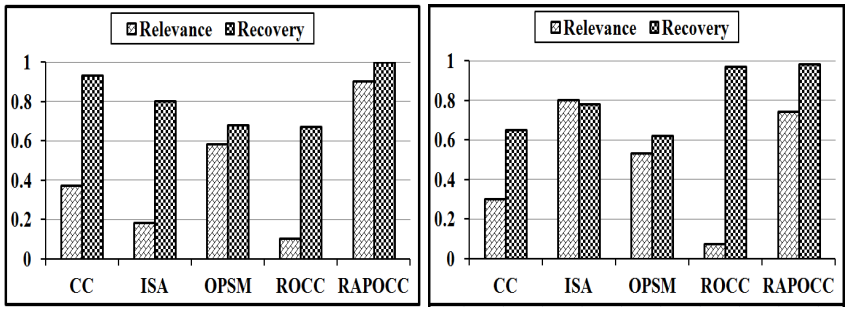
5.4.1 Experimental Setup

Datasets

For the synthetic datasets, a set of co-clusters were implanted in randomly generated datasets using the shifting and scaling patterns [143]. Given two rows, x and y, their relationship can be represented as:

$$x_j = y_j * s_{scale} + s_{shift}$$

where s_{shift} and s_{scale} are the shifting and scaling parameters. The sign of s_{shift} determines the correlation type: if $s_{shift} > 0$, then x and y are positively correlated, and if $s_{shift} < 0$, then x and y are negatively correlated [143]. In addition, two types of synthetic datasets were used, one without noise and the other with Gaussian noise. For the real-world datasets, we used eight expression datasets in the co-clustering experiments as described in Table 5.2.



(a) without noise (b) with 10% Gaussian noise

Figure 5.6: The co-clustering results on the synthetic datasets.

Comparisons with existing methods

In the co-clustering experiments, we compared the results of the *RAPOCC* algorithm against the *CC* [22], the OPSM [11], the *ISA* [64] and the *ROCC* [34] algorithms. We used BiCAT software (*http://www.tik.ethz.ch/sop/bicat/*) to run *CC*, *ISA* and *OPSM* algorithms using the default parameters. The code for the *ROCC* was obtained from the authors of [34].

Table 5.2: Description of the real-world gene expression datasets used in the co-clustering experiments

Dataset	**Genes**	**Samples**
Leukemia [48]	5000	38
Colon cancer [3]	2000	62
Medulloblastoma [91]	2059	23
Scleroderma [138]	2773	27
Arabidopsis thaliana [103]	734	69
Gasch yeast [103]	2993	173
Cho yeast [25]	6240	14
Causton yeast [19]	4960	11

Evaluation Measures

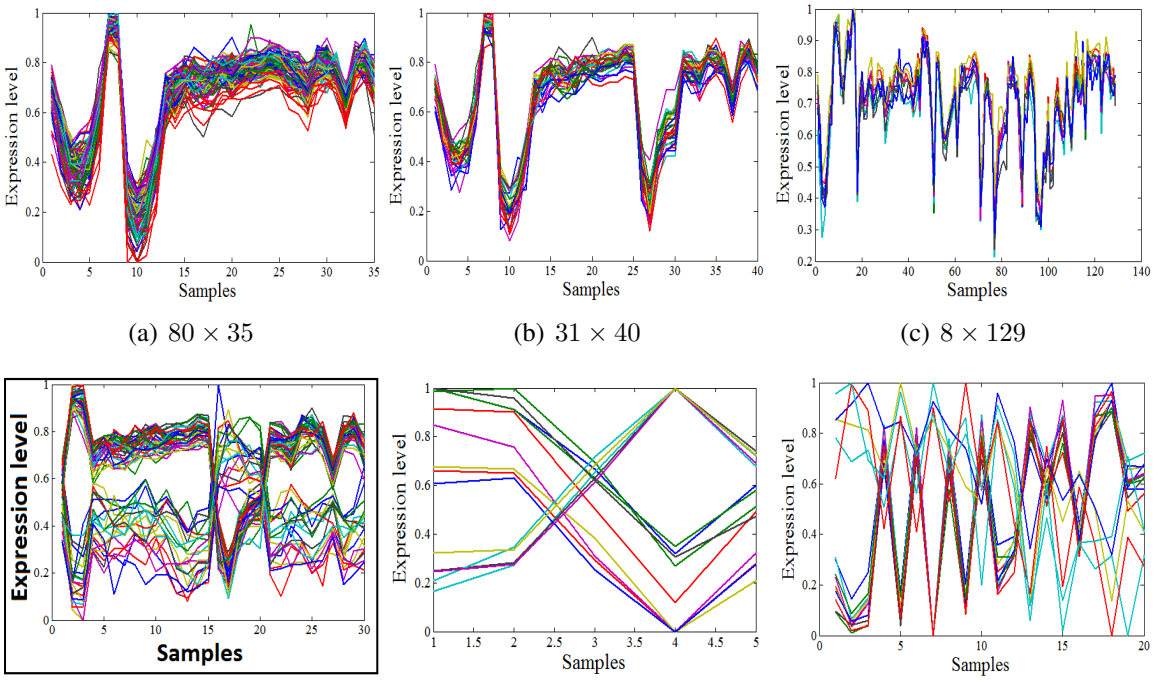
To evaluate the effectiveness of the proposed algorithm, we used several measures such as the **number of co-clusters**, **the average size** and the **average coherence** of the co-clusters computed using Definition 4. We also used the recovery and relevance measures [103]. **Recovery** determines how well each of the implanted co-clusters is discovered, and **relevance** is the extent to which the resulting co-clusters correspond the implanted co-clusters. Given a set of implanted co-clusters denoted by Y_{imp} and a set of co-clusters obtained by an algorithm denoted by X_{res}, the recovery and the relevance can defined as follows:

$$Recovery = \frac{1}{|Y_{imp}|} \sum_{(Y \in Y_{imp})} \arg\max_{(X \in X_{res})} \frac{|X \cap Y|}{|X \cup Y|}$$

$$Relevance = \frac{1}{|X_{res}|} \sum_{(X \in X_{res})} \arg\max_{(Y \in Y_{imp})} \frac{|X \cap Y|}{|X \cup Y|}$$

Biological Evaluation

The biological significance was estimated by calculating the p-values using the DAVID tool *(http://david.abcc.ncifcrf.gov/)* to test if a given co-cluster is enriched with genes from a



(a) 80×35 (b) 31×40 (c) 8×129

(d) 60×30 (e) 14×5 (f) 18×20

Figure 5.7: Examples of the co-clusters identified by the proposed RAPOCC algorithm on the gene expression datasets. The three co-clusters in the first row contain only the positively correlated genes which show similar patterns. The three co-clusters in the second row contain both positively and negatively correlated genes which show opposite patterns.

particular category to a greater extent than would be expected by chance [84]. When working with biological data, we are interested in identifying the biological significance of the results. The biological significance was estimated using the p-values with different significance levels = 5%, 1% and 0.1%. The hypergeometric distribution is used to calculate the probability of having at least k genes from a co-cluster of size n genes by chance in a biological process containing f genes from a total size of N genes as follows:

$$P = 1 - \sum_{i=0}^{k} \frac{\binom{f}{i}\binom{N-f}{n-i}}{\binom{N}{n}}$$

This test measures if a co-cluster is enriched with genes from a particular category to a greater extent than that would be expected by chance [84]. The range of the p-values is from 0 to 1. Lower p-values indicate biological significance [24].

Table 5.3: Results of the five co-clustering methods on the eight gene expression datasets

Dataset	Average coherence of co-clusters (No. of co-clusters)					Average volume of co-clusters (Avg No. of rows, Avg No. of columns)				
	CC	**ISA**	**OPSM**	**ROCC**	**RAPOCC**	**CC**	**ISA**	**OPSM**	**ROCC**	**RAPOCC**
Leukemia	0.9715 (20)	– (0)	0.9963 (37)	0.9775 (44)	**0.9984** (25)	7611.2 (310, 15)	–	**8475** (708, 20)	2544 (190, 10)	3543.7 (219, 13)
Colon	0.9884 (10)	0.9902 (21)	0.9810 (13)	0.9946 (62)	**0.9986** (11)	15.5 (5.9, 3.6)	376 (148.3, 5.6)	**2435** (619.2, 8.1)	881.2 (88, 10.6)	1437 (230.3, 7.8)
Medulloblastoma	0.9996 (10)	0.9906 (1)	0.9891 (10)	0.9892 (93)	**0.9997** (15)	16.6 (5.9, 6.5)	10 (5, 2)	**639** (225, 6)	258 (80.6, 3.2)	409.3 (82, 5)
Scleroderma	0.9838 (20)	0.9813 (2)	0.9862 (12)	0.9895 (47)	**0.9950** (20)	**2273.8** (110, 16)	15 (8, 2)	1303.4 (403, 8)	426 (63, 10)	1949 (380, 7)
Arabidopsis	0.9996 (20)	0.9569 (27)	0.9969 (12)	0.9952 (36)	**0.9998** (20)	146.2 (19, 8)	40.6 (20, 2)	330.7 (98, 8)	534.1 (41, 28)	**2282.1** (191, 12)
Gasch yeast	0.9844 (20)	0.9907 (63)	0.9966 (14)	0.9945 (87)	**0.9987** (25)	2424 (304, 43)	572.1 (67, 9)	2019.6 (522, 9)	2320.7 (115, 25)	**2582.5** (272, 29)
Cho yeast	0.9322 (20)	– (0)	0.9923 (11)	0.9854 (33)	**0.9960** (30)	950.5 (80, 12)	-	**2015** (682, 7)	757.9 (152, 6)	1958 (392, 5)
Causton yeast	0.9220 (17)	– (0)	0.9907 (9)	0.9831 (20)	**0.9965** (20)	2202.9 (219, 10)	-	2656.3 (941, 6)	800 (200, 4)	**3897.5** (780, 5)

5.4.2 Co-clustering Results

In this subsection, we present the results for the co-clustering experiments.

Results on Synthetic Data

Two types of datasets were used, one without noise and one with 10% noise. The size of each synthetic dataset is 200×150. Two co-clusters were implanted in each dataset, and the size of each co-cluster is 50×50. As shown in Figure 5.6, the *RAPOCC* algorithm outperformed the other algorithms because it optimizes for high-quality co-clusters. As a result, fewer random data points are added to the co-clusters obtained by our algorithm.

Results on Real Gene Expression Data

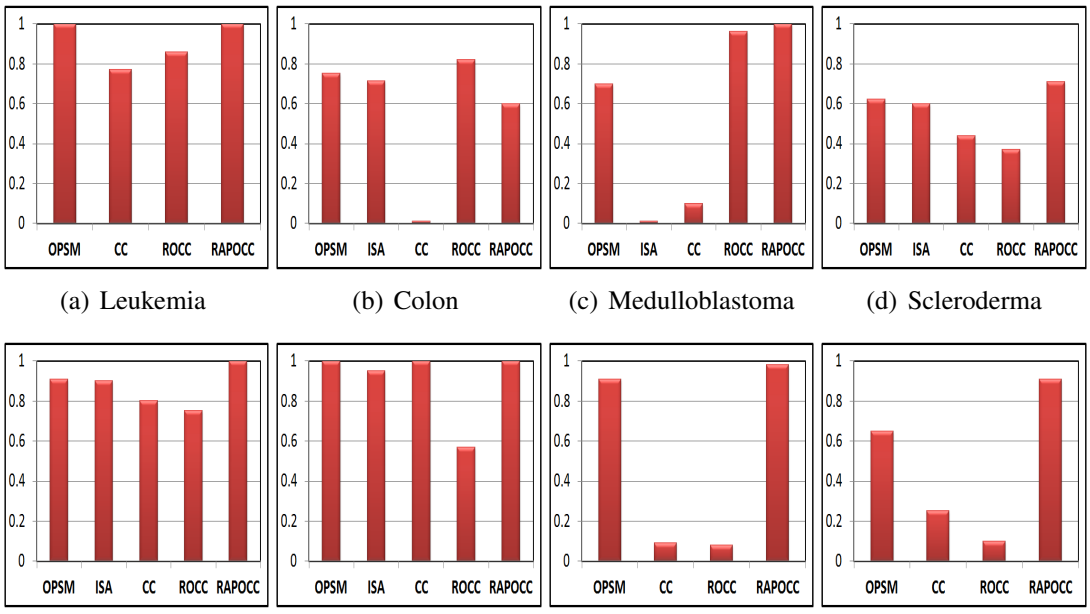
Figure 5.7 shows examples of the co-clusters identified by the proposed *RAPOCC* algorithm. The three co-clusters in the first row contain only the positively correlated genes which show similar patterns. These co-clusters were obtained from the Gasch yeast dataset. The three co-clusters in the second row contain both positively and negatively correlated genes which show opposite patterns. These co-clusters were obtained from Gash yeast, Scleroderma and Causton yeast datasets, respectively. The results of the five co-clustering methods on the eight datasets are shown in Table 5.3 and summarized in the following observations:

- **Coherence of the co-clusters.** The *RAPOCC* algorithm outperformed all the other algorithms on all of the datasets. The OPSM and the *ROCC* algorithms performed better than the CC and the *ISA* algorithms. These results confirmed one of our initial claims that *the proposed RAPOCC algorithm was designed to identify high-quality co-clusters*.

- **Size of the co-clusters.** Except for the *Leukemia* dataset, the *RAPOCC* produced either the largest or the second largest co-clusters in all of the datasets. The OPSM and the *RAPOCC* algorithms produced the largest co-clusters in four datasets and three datasets, respectively.

- **Number of the co-clusters.** The *ROCC* algorithm produced the largest number of co-clusters in all of the datasets. However, we observed that, in most of the cases, the co-clusters generated by this algorithm were either duplicates, subsets of each other or highly overlapping. On the other hand, the *ISA* algorithm did not produce any co-cluster for three datasets: *Leukemia*, *Cho yeast* and *Causton yeast*.

- **Biological significance of the co-clusters.** Figure 5.8 shows the average of the percentages of the biologically significant co-clusters using the DAVID tool from all the eight gene expression datasets. As shown in this figure, our proposed algorithm outperformed all other algorithms. The good performance of the OPSM algorithm in this context is due to the large size of co-clusters it generated.



(a) Leukemia (b) Colon (c) Medulloblastoma (d) Scleroderma

(e) Arabidopsis thaliana (f) Gasch yeast (g) Cho yeast (h) Causton yeast

Figure 5.8: Proportion of the co-clusters that are significantly enriched in each dataset (significance level = 5%).

In summary, the proposed co-clustering algorithm produced the higher quality, more biologically significant and relatively larger co-clusters compared to the other algorithms. Furthermore, the *RAPOCC* algorithm is more robust to noise.

5.5 Summary of the Co-clustering Algorithm

In this chapter, we presented a novel co-clustering algorithm (*RAPOCC*) to cluster large-scale gene expression data. It uses a novel objective function that is optimized to simultaneously find large co-clusters with minimum errors, and it allows positively and negatively correlated genes to be in the same co-cluster. The co-clusters can be arbitrarily positioned in the gene expression matrix and can overlap. Furthermore, the algorithm performs well on noisy data, and it can handle missing values. The experimental results on synthetic and real-word datasets showed that the proposed algorithm can extract biologically and statistically significant co-clusters from gene expression data. The proposed algorithm was compared to some of the existing algorithms, and the comparisons showed that the *RAPOCC* outperformed the other methods that are available in the literature.

CHAPTER 6

DIFFERENTIAL CO-CLUSTERING

6.1 Introduction

Discriminative models are used to analyze the differences between two classes and to identify class-specific patterns. Most of the existing discriminative models depend on using the entire feature space to compute the discriminative patterns for each class. Co-clustering has been proposed to capture the patterns that are correlated in a subset of features, but it cannot handle discriminative patterns in labeled datasets. In some applications, it is critical to consider the discriminative patterns that are correlated in a subset of the feature space. In this chapter, we extend the *RAPOCC* co-clustering algorithm to discover discriminative co-clusters by incorporating the class information into the co-cluster search process. In addition, we also characterize the discriminative co-clusters and propose three novel measures that can be used to evaluate the performance of any discriminative subspace algorithm. We evaluated the proposed algorithms on several synthetic and real gene expression datasets, and our experimental results showed that the proposed algorithms outperformed several existing algorithms available in the literature.

Discriminative models are used to extract patterns that are highly correlated in one class compared to another class. Mining such discriminative patterns can provide valuable knowledge toward understanding the differences between two classes and identifying class-specific patterns. For example, discriminative mining of gene expression data can lead to the identification of cancer-associated genes by comparing the expression patterns of the genes between healthy and cancerous tissues [32]. However, these genes can be correlated only in a subset of the cancerous samples due to the heterogeneity in the sample space [95]. Since the existing discriminative models are based on using all the features to find the discriminative patterns,

it is crucial to develop a model that can identify discriminative patterns that are correlated in a subset of the feature space. Figure 6.1 shows the correlations between three objects in two classes. These objects are highly correlated in a subset of the features in class A, but they are not correlated in class B. Such discriminative patterns cannot be discovered using standard discriminative models that use all the features. In order to capture these patterns, discriminative co-clustering is being proposed in this chapter.

Co-clustering has been proposed to identify subsets of objects that are inter-related under subsets of features (*co-clusters*) [22, 35, 34, 95, 146]. However, co-clustering is an unsupervised procedure that does not consider the class labels to find the discriminative patterns in labeled datasets. In order to capture the subspace discriminative patterns (or *discriminative co-clusters*), discriminative co-clustering is being proposed in this chapter by incorporating the class labels into the co-clustering process.

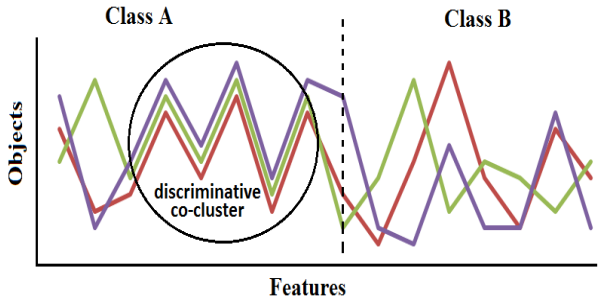


Figure 6.1: A set of three objects that are highly correlated in a subset of the features in class A, but they are not correlated in class B. Hence, these objects are considered as a discriminative (or differential) co-cluster.

6.1.1 Characteristics of Discriminative Co-clusters

Discriminative models aim to extract patterns that are differentially correlated between two classes [40]. In addition to the previously mentioned characteristics of the co-clusters, the discriminative co-clusters must possess the following characteristics:

1. **High discriminative coherence:** Coherence is a measure of similarity between a set of objects [95]. The discriminative co-clustering algorithms should identify the set of co-

clusters with the maximum difference in the coherence among the two classes. Trivial co-clusters that have the same correlation in both of the classes should be ignored.

2. **Low inter-class overlapping:** The discriminative co-clusters discovered in one class should have a minimal number of common rows with the co-clusters discovered in the other class.

3. **High discriminative power:** Incorporating the class labels can improve the performance of classification algorithms [61]. Discriminative co-clusters must be able to make more accurate predictions.

6.1.2 Motivating Example

Figure 6.2 shows an example of discriminative and non-discriminative co-clusters. The width of each co-cluster (X) indicates the number of features in it, and its shade represents its correlation score, which is also displayed as a percentage inside each co-cluster. The correlation score can be measured by various functions such as the mean-squared residue [22]. In this example, the higher the percentage (or the darker the shade), the stronger the correlation. The co-cluster properties (shade and width) are the main criteria used to distinguish between discriminative and non-discriminative co-clusters. A co-cluster is considered as a discriminative co-cluster if it is correlated only in one class (such as $X1$ and $X5.b$), if it is highly correlated in one class and less correlated in the other class (such as $X4$) or if it is correlated in relatively higher percentage of features (such as $X3$ and $X6$). The co-clusters $X2$ and $X5.a$ are not considered as discriminative co-clusters because they are similarly correlated in both classes. **Can any co-clustering algorithm be used to identify the discriminative co-clusters**? A naive solution to this problem is to co-cluster each class separately and then identify the co-clusters that appear in only one class. However, there are many limitations in following such a procedure: (i) Standard co-clustering algorithms focus on identifying the most correlated co-clusters. Therefore, discriminative co-clusters that have low correlation score (such as $X1$ and $X6$) will

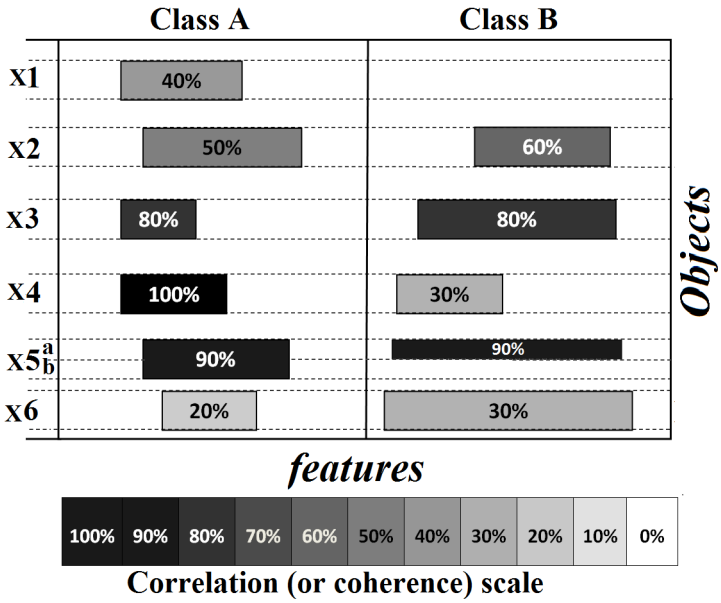


Figure 6.2: Example of discriminative co-clusters.

not be discovered. (ii) Since the standard co-clustering algorithms do not detect *all* the co-clusters, it is possible that co-cluster $X2$ is discovered only in one class and considered as a discriminative co-cluster. (iii) Most co-clustering algorithms prefer large co-clusters. Therefore, the complete co-cluster $X5$ may be considered as a discriminative co-cluster because part a is not discovered in class B due to its size limitation. *In this chapter, we develop a novel algorithm that directly optimizes an objective function to efficiently identify the discriminative co-clusters, and we propose two metrics to score the discriminative co-clusters based on their correlation scores and the number of features in them.*

6.1.3 Our Contributions

The purpose of this chapter is to present a novel discriminative co-clustering algorithm to efficiently find arbitrarily positioned co-clusters in the data matrix. The proposed algorithm can be used to discover discriminative co-clusters by incorporating the class information into the co-cluster discovery process. Our contributions can be summarized as follows:

1. Propose a novel discriminative co-clustering algorithm, **Di**scriminative **RAPOCC** (**Di-**

RAPOCC), to efficiently extract the discriminative co-clusters from labeled datasets.

2. Find the discriminative co-clusters from labeled datasets efficiently by incorporating the class information into the co-clustering process.

3. Propose three new evaluation metrics to quantify the results of the discriminative co-clustering algorithms on both synthetic and real gene expression datasets. Two metrics are used to measure the discriminative coherence property of the discriminative co-clusters, and the third one measures the inter-class overlap property.

4. Categorize the state-of-the-art approaches for discriminative co-clustering and characterize each category. We also empirically compare the performance of these categories with the proposed algorithm.

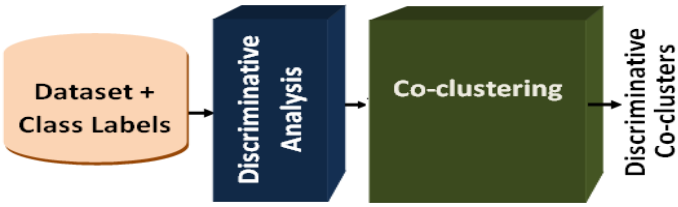
6.2 Differential Co-clustering Algorithms

In general, the co-clustering algorithms work in an unsupervised manner. However, some algorithms incorporate a priori knowledge in the co-clustering process. For example, in constrained co-clustering, some information can be incorporated such as the must-link and cannot-link constraints [102, 118, 114]. In discriminative co-clustering, the class labels are incorporated to find class-specific co-clusters. As illustrated in Figure 6.3, the existing discriminative co-clustering approaches can be categorized as two-step or one-step approaches.

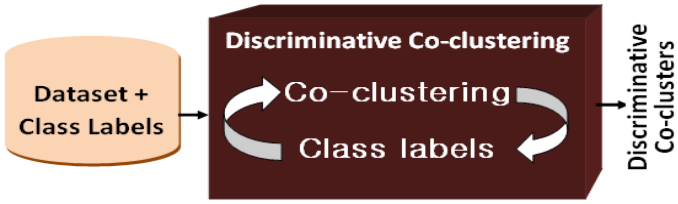
Two-step approaches

There are two sub-categories of these approaches: (*i*) **first co-clustering, and then discriminative analysis**. In [100], differentially expressed gene modules are identified by applying co-clustering each class separately, then the identified co-clusters are ranked based on their discrimination between the two classes. (*ii*) **first discriminative analysis, and then co-clustering**. The *DeBi* [112] algorithm uses two steps to identify differentially expressed

(a) Two-step approach



(b) Two-step approach



(c) The proposed model: one-step approach

Figure 6.3: Different approaches to obtain discriminative co-clusters.

co-clusters. The first step is to find the up or the down regulated genes using fold change analysis. In the second step, the *MAFIA* algorithm [15] is used to find the co-clusters from the up-regulation and the down-regulation data. There are two limitations for the two-step approaches: (i) the co-clustering is done for each class separately, and (ii) the discriminative analysis step is independent of the co-clustering step. Therefore, the one-step approaches have been proposed to overcome these limitations.

One-step approaches

The subspace differential co-expression (SDC) algorithm [39] uses the Apriori search algorithm to identify the discriminative patterns. The Apriori approach depends on using thresholds to define the discriminative patterns [39, 40]. For example, a given pattern is considered as a discriminative pattern if the difference between the correlations of this pattern in the two classes

is above a fixed threshold. Otherwise, this pattern will be split into smaller patterns to be tested again using the same threshold [39]. Therefore, the *SDC* method suffers from the following limitations: (i) It generates very small patterns [39]. (ii) The number of the discovered patterns dramatically grows with the size of the datasets, and it significantly varies with the threshold value [39, 40]. (iii) It has computational efficiency problems and does not scale well to large-scale datasets. In addition, the *SDC* method does not identify the subset of columns in which a given pattern shows the maximum correlation. In our previous work [99], we proposed a discriminative co-clustering algorithm to analyze the differences in the biological activities of several genes between two classes. Although this algorithm generated large co-clusters compared to the *SDC* method, this algorithm does not scale to large datasets because it maintains, for each pair of rows (genes), the set of columns under which the two rows are differentially correlated. Recently, locally discriminative co-clustering was proposed in [146] to explore the inter-sample and inter-feature relationships, but it does not find discriminative co-clusters as defined in our work. To overcome all of the above limitations of the existing approaches, we propose a novel discriminative co-clustering algorithm that directly optimizes an objective function to efficiently identify the discriminative co-clusters from a given labeled dataset.

6.3 The Proposed Differential Co-clustering Algorithm

6.3.1 Preliminaries and Problem Formulation

In this section, we introduce the coherence measure that can be used to measure the quality of the co-clusters, and we formulate the problems of co-clustering and discriminative co-clustering. The notations used in this chapter are described in Table 6.1, and we also used some of the notations from the previous chapter (Table 5.1). Here, we formally define the problems of co-clustering and discriminative co-clustering. Discriminative co-clustering aims to find the co-clusters that are highly correlated in one class but are less correlated in the other class. Mining discriminative co-clusters from labeled datasets is essential in several applications such as microarray data analysis and prediction.

Table 6.1: Notations used for the discriminative co-clustering algorithm.

Notation	Description
N^A	No. of columns in class A
K^A	number of optimized co-clusters in class A
c_j^A	j^{th} column in class A, $1 \leq j \leq \|N^A\|$
$X_k^A.r(i)$	i^{th} row of the k^{th} co-cluster in class A
$X_k^B.c(j)$	j^{th} column of the k^{th} co-cluster in class B

Definition 6. *(***Discriminative Co-clustering***). If $HP^A(X_i)$ measures the coherence of the co-cluster X_i in class A, the goal is to find the set of co-clusters that has maximal discriminative coherence*

$$\underset{X_1,X_2,\ldots,X_{K^A}}{\arg\max} \sum_{i=1}^{K^A} \left(HP^A(X_i) - \psi^B(X_i)\right)$$

$$\underset{X_1,X_2,\ldots,X_{K^B}}{\arg\max} \sum_{i=1}^{K^B} \left(HP^B(X_i) - \psi^A(X_i)\right)$$

where $\psi^A(X_i)$ ($\psi^B(X_i)$) is the maximum coherence of any subset of the objects in X_i in class A (B). The challenge here is to find $\psi(X_i)$, which is similar to the NP-hard problem of finding the maximum subspace in X_i [22]. In the proposed discriminative co-clustering algorithm, we propose two approximations for computing $\psi(X_i)$ that can be used to efficiently discover discriminative co-clusters by incorporating the class labels into the co-clusters discovery process.

Discriminative co-clustering aims to extract patterns that are highly correlated in a subset of the features in one class but not correlated in the other class. As illustrated in Figure 6.2, the rows of a discriminative co-cluster in one class should not form a co-cluster in the other class. This implies that there are two tasks that should be performed simultaneously: (i) search for a co-cluster in one class, and (ii) find the coherence of the rows of the co-cluster in the other class ($\psi^A(X)$ or $\psi^B(X)$ in Definition 6). **The challenge is to compute $\psi^B(X)$ ($\psi^A(X)$) while searching for the co-cluster in class A (B).**

Consider X^A as a co-cluster in class A that has $|I|$ rows and $|J^A|$ columns, and consider

$D^B(I,.)$ as the sub-matrix composed of the I rows and all the columns in class B. X^A will be considered as a discriminative co-cluster if there are no co-clusters in $D^B(I,.)$. An optimal solution for this would be to apply a co-clustering algorithm to find the maximal co-cluster in class $D^B(I,.)$. However, this is an NP-hard problem [22].

Table 6.2: A running example dataset for the discriminative co-clustering.

Row	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
x	9	8	6	5	3	2	1	6	8	5
y	10	4	10	6	9	3	2	9	9	10
z	2	5	8	4	5	9	8	9	1	8

An alternative solution to this problem is to consider the correlations of each pair of rows in $D^B(I,.)$. Given two rows (x and y) in $D^B(I,.)$, the aim is to find the subset of columns where the coherence between the two rows is maximized. To find an exact solution, one should enumerate all possible subsets of the $|N^B|$ columns. However, this solution is computationally infeasible since it requires enumerating all the $2^{|N^B|}$ subsets, where N^B is the number of columns in class B. To avoid such an exhaustive enumeration, we propose two efficient solutions: (i) a greedy-columns-selection solution and (ii) a clustering-based solution. Table 6.2 demonstrates a running example to illustrate how these solutions work.

6.3.2 Greedy-Columns-Selection

The intuition behind this measure is to iteratively compute the coherence between x and y based on the best J^i sets of columns for $1 \leq J^i \leq N^B$ and then report a weighted average of these N^B computations. In the first iteration, all the N^B columns are used. In the second iteration, one of the columns (j) is removed, and the remaining $N^B - 1$ columns are used to compute the coherence between the two rows. These are the set of $N^B - 1$ columns that achieves the maximum coherence between the two rows. This will be repeated to compute the coherence of the two rows using the best $N^B - 2, N^B - 3, ..., 1$ columns. The final value of

this measure is a weighted average of $\{h(x, y, J^1), ..., h(x, y, J^{N^B})\}$:

$$\frac{\sum_{i=1}^{N^B} h_+(x, y, J^i)|J^i|/N^B}{\sum_{i=1}^{N^B} |J^i|/N^B}$$

$$J^{(i+1)} = \{J^i\} - \arg\max_j \ h(x, y, \{J^i\} - \{j\})$$

$|J^i|/N^B$ is the weight assigned to each set of columns such that larger sets of columns are assigned more weight than smaller sets of columns. This measure can be used to capture the negative correlations by applying $h_-(x, y, J)$ instead of $h_+(x, y, J)$. Since no prior knowledge about the correlations between the rows is used, h_G will be computed twice, and the final value for this measure $h_G(x, y)$ is computed as the maximum of

$$\left(\frac{\sum_{i=1}^{N^B} h_+(x, y, J^i)|J^i|/N^B}{\sum_{i=1}^{N^B} |J^i|/N^B}, \frac{\sum_{i=1}^{N^B} h_-(x, y, J^i)|J^i|/N^B}{\sum_{i=1}^{N^B} |J^i|/N^B} \right)$$

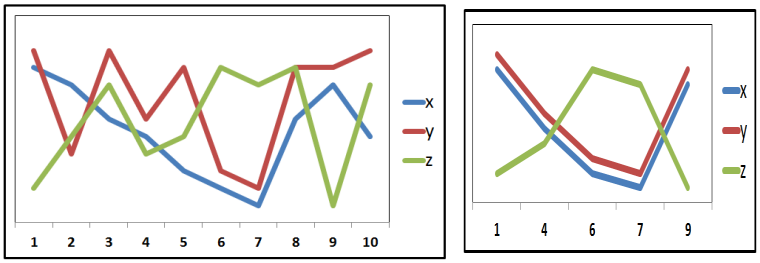
Finally, $\psi_G^B(X)$ is computed as:

$$\psi_G^B(X) = \frac{|2|}{|I|(|I| - 1)} \sum_{x,y \in X} h_G(x, y)$$

As an example, Table 6.3 shows the results of applying h_G on the x and y rows in Table 6.2. From this table, it should be noted that the two rows form a perfect co-cluster in the columns $\{c_1, c_4, c_6, c_7, c_9\}$. Figure 6.4(a) shows a plot for all the three rows in all the columns, and Figure 6.4(b) shows a plot for all the three rows in the identified subset of the columns. **Based on the greedy-columns-selection method, the first proposed discriminative coherence measure is defined as**

$$\Delta_G^A(X) = \psi_G^A(X) - \psi_G^B(X). \ \Delta_G^B(X) = \psi_G^B(X) - \psi_G^A(X)$$

The range of Δ_G^A and Δ_G^B is $(-1, 1)$.



(a) All columns. (b) A co-cluster.

Figure 6.4: (a) a plot for the entire running datasets. (b) a plot for the co-cluster extracted from the running dataset.

Table 6.3: Results of h_G on the x and y rows in Table 6.2.

(i)	Columns $\{J^m\}$	$h_+(x, y, J^i)$
1	$J^1 = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}\}$	0.9723
2	$J^2 = \{c_1, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}\}$	0.9860
3	$J^3 = \{c_1, c_3, c_4, c_6, c_7, c_8, c_9, c_{10}\}$	0.9908
4	$J^4 = \{c_1, c_3, c_4, c_6, c_7, c_8, c_9\}$	0.9947
5	$J^5 = \{c_1, c_4, c_6, c_7, c_8, c_9\}$	0.9978
6	$J^6 = \{c_1, c_4, c_6, c_7, c_9\}$	1.0
7	$J^7 = \{c_4, c_6, c_7, c_9\}$	1.0
8	$J^8 = \{c_6, c_7, c_9\}$	1.0
9	$J^9 = \{c_6, c_9\}$	1.0
10	$J^{10} = \{c_9\}$	1.0
$h_G(x, y, J)$ $(weighted\ average)$		**0.994**

6.3.3 Clustering-based discretization

The goal of the discretization step is to create a new representation of the data using a standard one-dimensional clustering algorithm to cluster each row separately. We rank the clusters in each row, and each value in a row will be represented by the rank of the cluster it belongs to. After clustering, we estimate the coherence between any two rows using the new representation.

The intuition of using clustering is to guarantee that similar data points within each row will be represented by the same value. The basic idea is as follows: (i) Cluster the values of each row to c clusters. (ii) Rank the clusters based on the mean of the values of each cluster such that cluster 1 contains the lowest values in x, and cluster c contains the highest values in

x. (iii) Map each value of x to the rank of the cluster the value belongs to.

$$\zeta : \{1, 2, ..., N^B\} \longrightarrow \{1, 2, ..., c\}$$

The positive correlation between two rows is defined as $(x_j - \bar{x}) = (y_j - \bar{y})$ and the negative correlation between them is defined as $(x_j - \bar{x}) = -(y_j - \bar{y})$. Using the new representation, the positive correlation can be represented as

$$\zeta(x_j) - \zeta(y_j) = s^+$$

where s^+ is the positive shift parameter. Since $\zeta(x_j)$ and $\zeta(y_j)$ can take any value between 1 and c , the shift parameter (s^+) can take any value from the following set: $\{-(c-1), ..., -1, 0, 1, ..., c-1\}$. Similarly, the negative correlation can be represented as

$$\zeta(x_j) + \zeta(y_j) = s^-$$

where s^- is the negative shift parameter that can take any value from the following set: $\{2, 3, ..., 2c\}$. Now, we can efficiently estimate the correlation between any two rows by finding the values of s^+ and s^- which will have a finite number of possible values. To estimate the positive correlation between x and y, we will subtract $\zeta(x_j)$ from $\zeta(y_j)$, and the most frequent value that appears in many columns will be considered as the value for s^+. Similarly, to estimate the negative correlation between x and y, we will add $\zeta(x_j)$ to $\zeta(y_j)$, and the most frequent value that appears in many columns will be considered as the value for s^-. To determine if the two rows are positively or negatively correlated, we compare the number of columns in which the two rows are considered positively correlated to the number of columns in which the two rows are considered negatively correlated.

$$J_{C^+} = \{j \mid \zeta(x_j) - \zeta(y_j) = s^+\}$$

$$J_{C^-} = \{j \mid \zeta(x_j) + \zeta(y_j) = s^-\}$$

If $|J_{C^+}| \geq |J_{C^-}|$, x and y are considered positively correlated, and their coherence is computed as $h_c(x,y) = h_+(x,y,J_{C^+})\frac{|J_{C^+}|}{|N^B|}$, else, x and y are considered negatively correlated, and their coherence is computed as $h_C(x,y) = h_-(x,y,J_{C^-})\frac{|J_{C^-}|}{|N^B|}$. Finally, ψ_C^B in class B can be computed as

$$\psi_C^B(X) = \frac{|2|}{|I|(|I|-1)} \sum_{x,y \in X} h_C(x,y)$$

To illustrate how this measure works, Table 6.4 shows the results of clustering each row in Table 6.2 (Here we used k-means, k=3. However, any other clustering algorithm can be used). The values in this table are the rankings of the clusters. For example, 1 indicates the cluster that has the lowest values in the corresponding row, and 3 indicates the cluster that has the maximum value. As an example, consider the first two rows. Subtracting $\zeta(x)$ from $\zeta(y)$ yields the following:

$$(0, 2, -1, 0, -2, 0, 0, -1, 0, -1)$$

This means that the maximum positive correlation between x and y is in 5 columns $\{c_1, c_4, c_6, c_7, c_9\}$ with $s^+ = 0$, while adding $\zeta(x)$ to $\zeta(y)$ yields

$$(6, 4, 5, 4, 4, 2, 2, 5, 6, 5)$$

This means that the maximum negative correlation between x and y is in 3 columns: $\{1, 4, 5\}$ with $s^- = 4$ or $\{c_3, c_8, c_{10}\}$ with $s^- = 5$). Hence, the coherence between x and y is computed as follows:

$$h_C(x,y) = h_+(x,y,\{c_1, c_4, c_6, c_7, c_9\})\frac{5}{10} = 0.5$$

As another example, the last two rows (y and z) are negatively correlated in the same set of columns:

$$h_C(y,z) = h_-\left(y,z,\{c_1, c_4, c_6, c_7, c_9\}\right)\frac{5}{10} = 0.5$$

The results here are similar to those obtained using h_G in terms of the set of columns in which the two rows have the maximum coherence, which is $\{c_1, c_4, c_6, c_7, c_9\}$. **Based on the clustering-based discretization method, the second proposed discriminative coherence measures is defined as follows:**

$$\Delta_C^A(X) = \psi_C^A(X) - \psi_C^B(X).\ \Delta_C^B(X) = \psi_C^B(X) - \psi_C^A(X)$$

Similar to Δ_G^A and Δ_G^B, the range of Δ_C^A and Δ_C^B is $(-1, 1)$. Our preliminary results showed that ψ_C and ψ_G produced very similar results on some of the simulated datasets. Since the computation of ψ_C is much faster than the computation of ψ_G, ψ_C is implemented in the proposed discriminative co-clustering algorithm. However, both measures will be used for evaluation purposes to quantify the resulting discriminative co-clusters using the proposed and the existing algorithms.

Table 6.4: Clustering of the running example dataset.

Row	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
$\zeta(x)$	3	3	2	2	1	1	1	2	3	2
$\zeta(y)$	3	1	3	2	3	1	1	3	3	3
$\zeta(z)$	1	2	3	2	2	3	3	3	1	3

6.3.4 The Di-RAPOCC Algorithm

The Di-RAPOCC algorithm, described in Algorithm 3, optimizes for the following objective function in order to extract the discriminative co-clusters.

Definition 7. *(***Discriminative Objective Function***) To obtain the top-K^A discriminative co-clusters from class A, the objective function can be written as:* $\arg\max_{X_1, X_2, \ldots, X_{K^A}} \sum_{i=1}^{K^A} \Phi^A(X)$ *where* $\Phi^A(X) = \big(HP^A(X_i) - \psi_C^B(X_i)\big)$. *To obtain the top-$K^B$ discriminative co-clusters from class B, the objective function can be written as:* $\arg\max_{X_1, X_2, \ldots, X_{K^B}} \sum_{i=1}^{K^B} \Phi^B(X)$ *where* $\Phi^B(X) = \big(HP^B(X_i) - \psi_C^A(X_i)\big)$.

Next, we will introduce all the steps of the proposed Di-RAPOCC algorithm.

Step 1: Initialize the K^A and K^B co-clusters. First, we compute h_C^A and h_C^B for all pairs of rows. This step is preceded by clustering the values of each class. The clustering is only used to identify the set of columns in which two rows have the maximum correlation, and the original values will be used in all the steps. Hence, there is no loss of information in this step. Then, we define $\delta_C^A(x, y)$ and $\delta_C^B(x, y)$ as follows:

$$\delta_C^A(x, y) = h_C^A(x, y) - h_C^B(x, y)$$

These will be used to identify K^A groups of rows, S^A, to be used as the seeds for the co-clusters (*lines 7-12*). If α is the minimum number of rows in any co-cluster, the candidate set for each row R_x is computed as follows:

$$R_x^A = \underset{r_1, r_2, \ldots, r_\alpha}{\arg\max} \sum_{i=1}^{\alpha} \delta_C^A(x, r_i) \tag{6.1}$$

From all of the M candidate sets (*since there are M rows in the data matrix, each row will be a candidate to be considered as a seed for a co-cluster*), the top-K^A sets are used as the initial co-clusters for each class.

$$S^A = \underset{S_1^A, S_2^A, \ldots, S_K^A}{\arg\max} \sum_{i=1}^{K^A} \left(\sum_{x,y \in R_i^A} \delta_C^A(x, y) \right) \tag{6.2}$$

Similarly, R^B and S^B will be computed for class B. Regarding the columns, all of them will be included in each co-cluster in the initialization.

Step 2: Updating the row/column clusterings. This is an iterative step in which we consider each row/column to be added/deleted from each co-cluster (*lines 13-27*). For each row, there are three possible assignments $\{-1, 0, 1\}$: 1 (-1) indicates adding the row to the co-cluster as positively (negatively) correlated, and 0 indicates removing the row from the corresponding

co-cluster. The assignments of the columns does not consider a negative correlation. The same row (or column) is allowed to be included in more than one co-cluster in this step. Similar to the *RAPOCC* algorithm, the convergence of the *Di-RAPOCC* algorithm is guaranteed since the maintained objective function is bounded and optimized to be monotonically increasing.

Algorithm 3 Di-RAPOCC(D, K, α),

Input: Data matrix (D)
No. of co-clusters (K^A and K^A)
Minimum No. of rows in any co-cluster (α)
Output:: A set of discriminative co-clusters ($\{X^A\}, \{X^B\}$)
Procedure:
Step 1: Compute δ_C for all the rows
$\forall x, y \in \{I\}\ \delta_C^A \leftarrow h_C^A(x,u) - h_C^B(x,u)$
$\forall x, y \in \{I\}\ \delta_C^B \leftarrow h_C^B(x,u) - h_C^A(x,u)$
Step 2: Initialize each of the K co-clusters for each class
Compute S^A and S^B *as defined in Section 5.3*
$/*$ Initialize rows and columns of each co-cluster $*/$
for $k = 1 : K$ **do**
  $\forall_{m \in S_k^A} X_k^A.r(m) = 1, \forall_{n \in N^A} X_k^A.c(n) = 1$
  $\forall_{m \in S_k^B} X_k^B.r(m) = 1, \forall_{n \in N^B} X_k^B.c(n) = 1$
end for
Step 3: Update the rows and the columns clusterings
repeat
  for $k = 1 : K$ **do**
    for $i = 1 : M$ **do**
      $X_k^A.r(i) = \arg\max_{u \in \{-1,0,1\}} \Phi(X_k^A.r(i) = u)$
      $X_k^B.r(i) = \arg\max_{u \in \{-1,0,1\}} \Phi(X_k^B.r(i) = u)$
    end for
    for $j = 1 : N^A$ **do**
      $X_k^A.c(j) = \arg\max_{v \in \{0,1\}} \Phi(X_k^A.c(j) = v)$
    end for
    for $j = 1 : N^B$ **do**
      $X_k^B.c(j) = \arg\max_{v \in \{0,1\}} \Phi(X_k^B.c(j) = v)$
    end for
  end for
until convergence
Step 3: Merging similar co-clusters.
Step 4: Pruning.
return ($\{X^A\}, \{X^B\}$)

Step 3: Merging the Co-clusters. Similar to the *RAPOCC* algorithm, the goal of this step is to merge similar co-clusters using an agglomerative clustering approach. The two most

similar co-clusters, within the same class, are merged in each iteration. This step allows the discovery of **large** discriminative co-clusters, and it allows **intra-class overlapping** co-clusters.

Step 4: Pruning. In this step, we prune the co-clusters with the lowest discriminative scores. To determine which co-clusters to prune, (i) sort the co-clusters based on $\Phi^A(X)$, in class A and $\Phi^B(X)$, in class B, (ii) compute the difference between the consecutive scores and (iii) report the set of co-clusters just before the largest difference, and prune the remaining co-clusters.

6.4 The Experimental Results

To demonstrate the effectiveness of the proposed algorithms, several experiments were conducted using both synthetic and real-world gene expression datasets.

6.4.1 Experimental Setup

Datastes

For the synthetic datasets, a set of co-clusters were implanted in randomly generated datasets using the shifting and scaling patterns [143]. In addition, two types of synthetic datasets were used, one without noise and the other with Gaussian noise. For the real-world datasets, we used the four gene expression datasets.

Comparisons with existing methods

In the discriminative co-clustering experiments, we compared the results of the *Di-RAPOCC* algorithm against the SDC algorithm [39] and the *OPSM* algorithm [11]. The *OPSM* algorithm is not a discriminative co-clustering algorithm. Therefore, we used the following procedure: (*i*) Apply OPSM on each class separately, (*ii*) compute the inter-class overlap, (*iii*) remove the co-clusters that have inter-class overlap $\geq 50\%$, and *(iv)* report the remaining co-clusters. We refer to this modified algorithm as Discriminative OPSM (*Di-OPSM*). The SDC algorithm takes as input three parameters $(SDC, r, minpattsize)$ [39], which were set to the default val-

ues: $(0.2, 0.2, 3)$ unless otherwise stated.

Evaluation Measures

In addition to the co-clustering evaluation measures presented in the previous chapter Section 5.4 (number of co-clusters, the average size, average coherence, recovery and relevance), we used the following proposed metrics to evaluate the results of the discriminative co-clustering:

- **Greedy-based discriminative coherence (Δ_G)**

$$\Delta_G = \frac{1}{(K^A + K^B)} \left(\sum_{k=1}^{K^A} \Delta_G^A + \sum_{k=1}^{K^B} \Delta_G^B \right)$$

- **Clustering-based discriminative coherence (Δ_C)**

$$\Delta_C = \frac{1}{(K^A + K^B)} \left(\sum_{k=1}^{K^A} \Delta_C^A + \sum_{k=1}^{K^B} \Delta_C^B \right)$$

- **Inter-class overlap.** If X^A (X^B) is the set of discriminative co-clusters in class A (B), the inter-class overlap is defined as the average of:

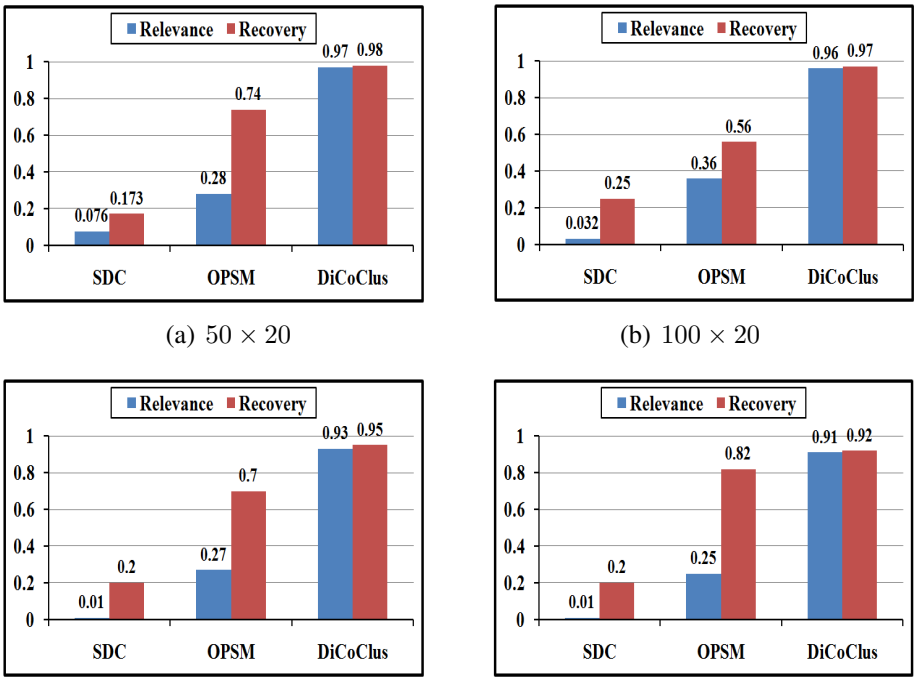
$$\left(\sum_{k=1}^{K^A} \underset{X_k^B}{\arg\max} \frac{|X_k^A \cap X_k^B|}{|X_k^A \cup X_k^B|} + \sum_{k=1}^{K^B} \underset{X_k^A}{\arg\max} \frac{|X_k^B \cap X_k^A|}{|X_{k^B}^B \cup X_k^A|} \right)$$

  where the union and intersection operations are computed using the rows in the co-clusters.

The biological significance was estimated by calculating the p-values using the DAVID tool as described in Section 5.4.

6.4.2 Differential Co-clustering Results

In this subsection, we present the results for discriminative co-clustering experiments. Due to space limitations, in some of the tables we used *OPM* and *RPC* to refer to *Di-OPSM* and



(a) 50×20 (b) 100×20

(c) 300×20 (d) 500×20

Figure 6.5: Relevance and Recovery for SDC, OPSM and DiCoClus, respectively obtained from different synthetic datasets.

Di-RAPOCC algorithms, respectively.

Table 6.5: Number of co-clusters from synthetic datasets.

Synthetic dataset	SDC	Di-OPSM	Di-RAPOCC
s= 50	256	15	2
s= 100	990	16	2
s= 300	4, 451	16	3
s= 500	10, 210	22	3

Results on Synthetic Data

Using the shifting-and-scaling model [143], four co-clusters were generated of the size 10×10. Half of those co-clusters were designed to be discriminative, while the remaining co-clusters were common in both classes. The structure of the synthetic datasets is similar

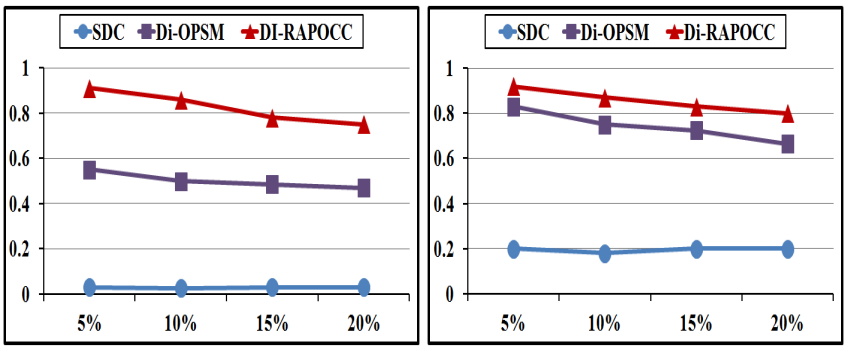
Table 6.6: Discriminative measures (synthetic datasets).

Synthetic dataset	Δ_G			Δ_C		
	SDC	OPM	RPC	SDC	OPM	RPC
s=50, η=0	0.51	0.54	**0.69**	0.51	0.55	**0.72**
s=100, η=0	0.50	0.68	**0.71**	0.54	0.0.54	**0.70**
s=200, η=0	0.49	0.63	**0.70**	0.54	0.66	**0.71**
s=300, η=0	0.52	0.51	**0.67**	0.51	0.64	**0.70**
s=500, η=0	0.51	0.64	**0.71**	0.52	0.63	**0.72**
s=100, η=5%	0.53	0.57	**0.71**	0.51	0.60	**0.70**
s=100, η=10%	0.52	0.65	**0.67**	0.53	0.61	**0.65**
s=100, η=15%	0.51	0.63	**0.76**	0.49	0.63	**0.70**
s=100, η=20%	0.52	0.64	**0.72**	0.50	0.61	**0.65**

Table 6.7: Description of the real-world gene expression datasets used in the differential co-clustering experiments

Dataset	**Genes**	**Total samples**	**class A**		**class B**	
			Description	**samples**	**Description**	**samples**
Leukemia [48]	5000	38	Acute lymphoblastic leukemia	11	Acute myeloid leukemia	27
Colon cancer [3]	2000	62	Normal	22	Tumor	40
Medulloblastoma [91]	2059	23	Metastatic	10	Non-metastatic	13
Scleroderma [138]	2773	27	Male	12	Female	15

to the structure shown in Figure 6.2. In the first experiment, we implanted the synthetic co-clusters in random matrices of different sizes given by $s \times 20$, where $s = (50, 100, 300, 500)$. Figure 6.5 shows the relevance and recovery results of *SDC*, *Di-OPSM* and *Di-RAPOCC* co-clustering algorithms when applied to the synthetic datasets. The noise level, η, in this set of experiments is 0. The proposed algorithm outperformed other algorithms indicating that the proposed algorithm is capable of identifying the discriminative co-clusters. Since *Di-OPSM* was not directly designed to extract discriminative co-clusters, the identified co-clusters include both discriminative and non-discriminative co-clusters. The poor performance of the *SDC* algorithm can be explained by two main reasons. (i) *SDC* generates too many patterns as shown in Table 6.5. As the size of the dataset increases, the number of the generated patterns generated by the *SDC* algorithm *increases dramatically*. (ii) The *SDC* algorithm generates *very small patterns* (average of 3 rows per pattern). On the other hand, the *Di-RAPOCC* algorithm prunes any non-discriminative co-cluster.



(a) Relevance (b) Recovery

Figure 6.6: Relevance and recovery obtained with noise levels of 5%, 10%, 15% and 20%, respectively.

Table 6.8: Discriminative measures (expression datasets).

Dataset	Δ_G			Δ_C		
	SDC	OPM	RPC	SDC	OPM	RPC
Colon	0.60	0.58	**0.62**	0.50	0.53	**0.56**
Medulloblastoma	0.49	0.54	**0.59**	0.51	0.53	**0.55**
Leukemia	-	0.57	**0.59**	-	0.56	**0.58**
Scleroderma	0.57	0.54	**0.60**	0.54	0.55	**0.60**

In the second experiment, different levels of noise were used, which are $0, 5\%, 10\%, 15\%$ and 20%, respectively, to the synthetic dataset of size 100×20. Figure 6.6 shows the recovery and the relevance of the three algorithms. As the noise level increases in the dataset, the relevance and the recovery values are degraded. However, our algorithm is still the algorithm most robust to noise due to the use of a clustering approach to estimate the coherence of any co-cluster. Table 6.6 shows the average results of the discriminative measurements Δ_G and Δ_C for all the different synthetic datasets. Unsurprisingly, our algorithm achieved the best results in all the datasets because it primarily focuses on identifying the most discriminative co-clusters in the search process. Figure 6.7 shows the inter-class overlap on synthetic datasets. The *Di-RAPOCC* algorithm achieved the best results because it avoids common patterns in both of the classes.

Table 6.9: Results of differential co-clustering.

Dataset	No. of co-clusters in A			No. of co-clusters in B			Overlap			Average coherence (H)	
	SDC	OPM	RPC	SDC	OPM	RPC	SDC	OPM	RPC	OPM	RPC
Colon	155	10	15	1	3	13	**0.0**	0.01	0.04	0.992	**0.997**
Medulloblastoma	74,957	8	14	7,597	9	14	0.2	0.12	**0.01**	0.988	**0.994**
Leukemia	-	21	35	-	5	22	-	0.40	**0.09**	0.990	**0.995**
Scleroderma	48,623	12	10	469	10	9	0.04	0.17	**0.0**	0.986	**0.998**

Results on Real Gene Expression Data

For the real-world datasets, we used the four datasets as described in Table 6.7. Each dataset has two distinct classes of biological samples. The SDC algorithm was applied on the *Medulloblastoma* and the *Scleroderma* datasets with the parameters values set to $(0.3, 0.3, 3)$ to avoid *out of memory* problems. For the *Leukemia* datasets, out of memory errors occurred for different combinations of the parameters; therefore, there are no results for this dataset. As shown in Table 6.8, the *Di-RAPOCC* algorithm achieved the best results in terms of the discriminative coherence measures (Δ_G and Δ_C). The results were also analyzed in terms of the number of co-clusters, the inter-class overlap and the average coherence as shown in Table 6.9. The coherence measure cannot be applied to the results of the SDC algorithm because it does not report the columns in which a set of rows is correlated. Here, we make some remarks regarding the performance of the three algorithms.

Table 6.10: Comparisons between the three differential co-clustering algorithms.

Measure	**SDC**	**Di-OPSM**	**Di-RARPOCC**
No. of the co-clusters	High	Low	Medium
Size of the co-clusters	Small	Large	Medium
Coherence	-	Low	High
Discriminative coherence	Low	Medium	High
Inter-class overlap	High	Medium	Low
Recovery	Low	Medium	High
Relevance	Low	Medium	High

- The **SDC** algorithm tends to produce a large number of small patterns. Since the *SDC*

algorithm uses the Apriori approach, it has some computational efficiency problems, and the number of the discovered patterns grows dramatically with larger datasets.

- The **Di-OPSM** algorithm tends to produce co-clusters that are too large. Therefore, it does not give good results in terms of the coherence, inter-class overlap and discriminative measures. Since it is not a discriminative co-clustering algorithm, we have to run it on each class independently.

- The **Di-RAPOCC** algorithm keeps the top discriminative co-clusters and prunes the other co-clusters, and it works well on noisy and large datasets.

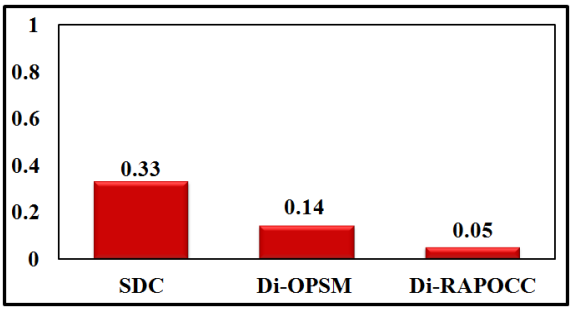


Figure 6.7: The inter-class overlapping on synthetic datasets.

Figure 6.8 shows the biological evaluation of the results. The *SDC* algorithm was excluded from this analysis because it produced too many patterns. The *Di-RAPOCC* algorithm outperformed the *Di-OPSM* algorithm in three datasets, while OPSM was better in the *Leukemia* dataset. However, for this dataset, *Di-RAPOCC* outperformed *Di-OPSM* in terms of the inter-class overlap, the coherence and the discriminative coherence measures. In a different analysis, we found several significant biological pathways that were enriched in the co-clusters produced by the proposed algorithm. For example, the *MAPK signaling pathway* which has a p-value = $4.77E-12$ was reported as an up-regulated pathway in the metastatic tumors that is very relevant to the study of metastatic disease [91]. The summary of comparisons between the three algorithms is shown in Table 6.10.

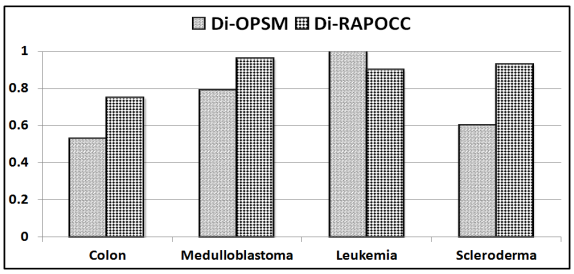


Figure 6.8: Proportion of the discriminative co-clusters that are significantly enriched in each dataset (significance level = 5%).

6.5 Summary of the Differential Co-clustering Algorithm

In this chapter, we presented a novel algorithm for discovering discriminative co-clusters. The proposed algorithm integrates the class label in the co-clustering discovery process, and it works well on noisy datasets. The experimental results showed that the proposed algorithm outperforms the existing algorithms and can extract biologically and statistically significant discriminative co-clusters. As a future work, we are interested in analyzing its discriminative power of the proposed approach and extending it to solve prediction problems.

CHAPTER 7

CONCLUSION AND FUTURE WORK

Understanding the mechanisms of cancer and other diseases requires analyzing the differences between the two phenotypes normal (or control) and cancerous (or treated). Most of the existing computational approaches depend on testing the changes in the expression levels of each single gene individually. In this work, we proposed novel computational approaches to find the differential genes between two phenotypes. The proposed approaches are grouped as: differential network analysis and differential co-clustering. The proposed models can quantitatively and qualitatively characterize the differences between two classes (or two phenotypes) and can provide better insights and understandings of various diseases.

The goal of the first proposed approach is to represent the two phenotypes as two networks, and then the problem of identifying differential genes is transformed to the problem of comparing two networks to identify the most differential network components. Networks have been extensively used to model various complex systems such as online social networks and biological networks. Studying such networks can provide valuable knowledge about the data objects and their interactions. Therefore, we proposed two novel differential networking algorithms to identify differential hubs and differential subnetworks, respectively. The first differential network algorithm is called the *DiffRank* algorithm, which ranks the nodes of two networks based on their differential behavior. We defined novel differential measures such as differential connectivity and differential centrality for each node. These measures are propagated through the network and are optimized to capture the local and global structural changes between two networks. We demonstrated the effectiveness of *DiffRank* on synthetic datasets and real-world applications and showed tha *DiffRank* identifies meaningful and valuable information compared to some of the baseline methods that can be used for such a task.

The*DiffRank* algorithm has two salient features. First, it can effectively capture the differences in both local and global structures between two networks. Second, it iteratively propagate the novel differential scores through the network until convergence to obtain accurate rankings for all the nodes. Therefore, we integrated the results of the *DiffRank* algorithm in the proposed differential subnetwork algorithm which is called *DiffSubNet*. This algorithm aims to identify sets of differentially connected nodes. Motivated by the guilt-by-association principle which states that genes with similar functions exhibit similar expression patterns (co-expressed) [132, 33], we proposed a novel network-based differential subnetwork algorithm to identify differential subnetworks between two networks. The differential subnetworks are groups of strongly connected nodes in one network but not in the other.

The major limitation in the proposed networking-based algorithms is its sensitivity to the network construction method. The *DiffRank* and the *DiffSubNet* algorithms take as input two networks. If the networks are not pre-defined, we need to construct them from the raw data. Hence, using different network construction methods with different parameters will yield different results. To resolve this issue, we recommend to integrate prior knowledge and the domain experts to guide the process of network construction. However, in other domains, the networks are already predefined. This include PPI networks and social networks.

The goal of the second approach is to discover a distinguishing set of gene patterns that are highly correlated in a subset of the samples in one phenotype but not in the other. This approach is useful when the biological samples are assumed to be heterogenous or have multiple subtypes where a set of genes can be co-expressed only in a subset of the samples (subspace co-expression). The unique characteristic of the proposed differential co-clustering algorithm is that it incorporates the class labels of the data in the co-clustering process. co-clustering is an unsupervised learning process, but our proposed approach aims to find class-specific patterns by integrating the class labels in the search process. The extensive experimental results showed that the proposed algorithm outperforms the existing algorithms and can extract biologically

and statistically significant discriminative co-clusters from synthetic datasets and real-world datasets.

The main challenge in the co-clustering-based approach is how to predefine the optimal number of co-clusters for a given dataset. This is a common problem in all the clustering algorithms. In our proposed co-clustering approach, we added two operations to minimize the effects of this problem. These operations are merging similar co-clusters and pruning the irrelevant co-clusters. In addition, our approach ranks the resulting co-clusters to enable the biologist to focus on a small subset of them that capture the differences between the two phenotypes under study.

Our work opens the door to several interesting directions for future work. Mainly, we are interested in: (i) Analyzing its discriminative power of the proposed approaches and extending them to solve prediction problems. (ii) Applying the proposed approaches in other domains.

- **Solving prediction problem**. The differential patterns discovered by the proposed framework can be used as predictive patterns. Since these patterns are identified based on the differences between the two biological conditions, they can be used to discriminate between the two phenotypes. Mining such discriminative patterns can provide valuable knowledge toward understanding the differences between two classes and identifying class-specific patterns. The proposed approaches generate three types of discriminative patterns: differential hubs, differential subnetworks and differential co-clusters. Since incorporating the class labels can improve the performance of classification algorithms, these discriminative patterns must be able to make more accurate predictions. In our work, we have focused on how to efficiently identifying the discriminative patterns from gene expression data. In the future, we are interested in investigating the discriminative power of these patterns and integrating them in prediction systems.

- **Considering other domains**. We have focused on gene expression data as the main application of our work. One of the main advantages of our novel approaches is that

they can be applied to solve various problems that depend on comparing two classes. In addition to the phenotypic variation, there are several other sources of variation such as temporal and topic variations which can be modeled as two-classes problems. Here, we can use the differential networking algorithms to model this problem and find interesting network components that are relevant to change over time.

Another interesting future study is to further explore the problem of differential networking analysis in heterogenous or multi-mode networks. In addition, one can investigate how to integrate the concepts of influential nodes [135] and effectors [79] in the differential analysis of multiple social networks.

BIBLIOGRAPHY

[1] ACKERMANN, M., AND STRIMMER, K. A general modular framework for gene set enrichment analysis. *BMC Bioinformatics 10*, 1 (2009), 47.

[2] ALAYDIE, N., AND FOTOUHI, F. Unraveling complex relationships between heterogeneous omics datasets using local principal components. In *IRI* (2011), IEEE Systems, Man, and Cybernetics Society, pp. 136–141.

[3] ALON, U., BARKAI, N., NOTTERMAN, D. A., GISH, K., YBARRA, S., MACK, D., AND LEVINE, A. J. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America 96*, 12 (June 1999), 6745–6750.

[4] ANAGNOSTOPOULOS, A., DASGUPTA, A., AND KUMAR, R. Approximation algorithms for co-clustering. In *PODS '08: Proceedings of the twenty-seventh ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (New York, NY, USA, 2008), pp. 201–210.

[5] BANDYOPADHYAY, S., MEHTA, M., KUO, D., SUNG, M.-K., CHUANG, R., JAEHNIG, E. J., BODENMILLER, B., LICON, K., COPELAND, W., SHALES, M., FIEDLER, D., DUTKOWSKI, J., GUNOL, A., VAN ATTIKUM, H., SHOKAT, K. M., KOLODNER, R. D., HUH, W.-K., AEBERSOLD, R., KEOGH, M.-C., KROGAN, N. J., AND IDEKER, T. Rewiring of genetic networks in response to dna damage. *Science 330*, 6009 (2010), 1385–1389.

[6] BANERJEE, A., DHILLON, I., GHOSH, J., MERUGU, S., AND MODHA, D. S. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. *J. Mach. Learn. Res. 8* (2007), 1919–1986.

[7] BARABASI, A.-L., AND ALBERT, R. Emergence of Scaling in Random Networks. *Science 286*, 5439 (1999), 509–512.

[8] BARABASI, A.-L., AND OLTVAI, Z. N. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics 5*, 2 (Feb. 2004), 101–113.

[9] BARRY, W. T., NOBEL, A. B., AND WRIGHT, F. A. Significance analysis of functional categories in gene expression studies: a structured permutation approach. *Bioinformatics 21*, 9 (2005), 1943–1949.

[10] BECKERS, J., HERRMANN, F., RIEGER, S., DROBYSHEV, A. L., HORSCH, M., HRABEACUTE; DE ANGELIS, M., AND SELIGER, B. Identification and validation of novel erbb2 (her2, neu) targets including genes involved in angiogenesis. *Int J Cancer 114*, 4 (2005), 590–7.

[11] BEN-DOR, A., CHOR, B., KARP, R., AND YAKHINI, Z. Discovering local structure in gene expression data: the order-preserving submatrix problem. *Journal of computational biology 10*, 3-4 (2003), 373–384.

[12] BRAUN, R., COPE, L., AND PARMIGIANI, G. Identifying differential correlation in gene/pathway combinations. *BMC Bioinformatics 9*, 1 (2008), 488.

[13] BRIN, S., AND PAGE, L. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web 7* (1998), WWW7, pp. 107–117.

[14] BROBERG, P. Ranking genes with respect to differential expression. *Genome Biology 3*, 9 (2002), preprint0007.1–preprint0007.23.

[15] BURDICK, D. MAFIA: A maximal frequent itemset algorithm for transactional databases. In *Proceedings of the 17th International Conference on Data Engineering* (2001), ICDE '01, pp. 443–452–.

[16] BUSYGIN, S., PROKOPYEV, O., AND PARDALOS, P. M. Biclustering in data mining. *Comput. Oper. Res. 35*, 9 (2008), 2964–2987.

[17] CABUSORA, L., SUTTON, E., FULMER, A., AND FORST, C. V. Differential network expression during drug and stress response. *Bioinformatics 21* (2005), 2898–2905.

[18] CARTER, S. L., BRECHBHLER, C. M., GRIFFIN, M., AND BOND, A. T. Gene co-expression network topology provides a framework for molecular characterization of cellular state. *Bioinformatics 20*, 14 (2004), 2242–2250.

[19] CAUSTON, H. C., REN, B., KOH, S. S., HARBISON, C. T., KANIN, E., JENNINGS, E. G., LEE, T. I., TRUE, H. L., LANDER, E. S., AND YOUNG, R. A. Remodeling of yeast genome expression in response to environmental changes. *Mol Biol Cell 12*, 2 (2001), 323–337.

[20] CHEN, C., YAN, X., ZHU, F., HAN, J., AND YU, P. S. Graph OLAP: Towards online analytical processing on graphs. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining* (Washington, DC, USA, 2008), pp. 103–112.

[21] CHEN, W., WANG, C., AND WANG, Y. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2010), KDD '10, pp. 1029–1038.

[22] CHENG, Y., AND CHURCH, G. M. Biclustering of expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology* (2000), AAAI Press, pp. 93–103.

[23] CHEUNG, V. G., CONLIN, L. K., WEBER, T. M., ARCARO, M., JEN, K.-Y., MORLEY, M., AND SPIELMAN, R. S. Natural variation in human gene expression assessed in lymphoblastoid cells. *Nature Genetics 33*, 3 (Feb. 2003), 422–425.

[24] CHO, H., AND DHILLON, I. S. Coclustering of human cancer microarrays using minimum sum-squared residue coclustering. *IEEE/ACM Trans. Comput. Biol. Bioinformatics 5*, 3 (2008), 385–400.

[25] CHO, R. J., CAMPBELL, M. J., WINZELER, E. A., STEINMETZ, L., CONWAY, A., WODICKA, L., WOLFSBERG, T. G., GABRIELIAN, A. E., LANDSMAN, D., LOCKHART, D. J., AND DAVIS, R. W. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular cell 2*, 1 (1998), 65–73.

[26] CHO, S., KIM, J., AND KIM, J. Identifying set-wise differential co-expression in gene expression microarray data. *BMC Bioinformatics 10*, 1 (2009), 109.

[27] CHOI, J. K., YU, U., YOO, O. J., AND KIM, S. Differential coexpression analysis using microarray data and its application to human cancer. *Bioinformatics 21*, 24 (2005), 4348–4355.

[28] CHOI, Y., AND KENDZIORSKI, C. Statistical methods for gene set co-expression analysis. *Bioinformatics 25*, 21 (2009), 2780–2786.

[29] CHOWDHURY, S., NIBBE, R., CHANCE, M., AND KOYUTÜRK, M. Subnetwork State Functions Define Dysregulated Subnetworks in Cancer. In *Research in Computational Molecular Biology*, B. Berger, Ed., vol. 6044 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2010, ch. 6, pp. 80–95.

[30] CHU, J.-H., LAZARUS, R., CAREY, V., AND RABY, B. Quantifying differential gene connectivity between disease states for objective identification of disease-relevant genes. *BMC Systems Biology 5*, 1 (2011), 89.

[31] DAO, P., WANG, K., COLLINS, C., ESTER, M., LAPUK, A., AND SAHINALP, S. C. Optimally discriminative subnetwork markers predict response to chemotherapy. *Bioinformatics 27*, 13 (2011), i205–i213.

[32] DE LA FUENTE, A. From 'differential expression' to 'differential networking' identification of dysfunctional regulatory networks in diseases. *Trends in Genetics 26*, 7 (July 2010), 326–333.

[33] DE SMET, R., AND MARCHAL, K. Advantages and limitations of current network inference methods. *Nat Rev Micro 8*, 10 (Oct. 2010), 717–729.

[34] DEODHAR, M., GUPTA, G., GHOSH, J., CHO, H., AND DHILLON, I. A scalable framework for discovering coherent co-clusters in noisy data. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning* (2009), pp. 241–248.

[35] DHILLON, I. S., MALLELA, S., AND MODHA, D. S. Information-theoretic co-clustering. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2003), ACM, pp. 89–98.

[36] DING, Y., YAN, E., FRAZHO, A., AND CAVERLEE, J. Pagerank for ranking authors in co-citation networks. *J. Am. Soc. Inf. Sci. Technol. 60* (November 2009), 2229–2243.

[37] EFRON, B., AND TIBSHIRANI, R. On testing the significance of sets of genes. *Annals of Applied Statistics 1* (2007), 107–120.

[38] ELO, L. L., JRVENP, H., OREIC, M., LAHESMAA, R., AND AITTOKALLIO, T. Systematic construction of gene coexpression networks with applications to human t helper cell differentiation process. *Bioinformatics 23*, 16 (2007), 2096–2103.

[39] FANG, G., KUANG, R., PANDEY, G., STEINBACH, M., MYERS, C. L., AND KUMAR, V. Subspace differential coexpression analysis: problem definition and a general approach. *Pacific Symposium on Biocomputing* (2010), 145–156.

[40] FANG, G., PANDEY, G., WANG, W., GUPTA, M., STEINBACH, M., AND KUMAR, V. Mining low-support discriminative patterns from dense and high-dimensional data. *IEEE Transactions on Knowledge and Data Engineering 24* (2012), 279–294.

[41] FRANCESCONI, M., REMONDINI, D., NERETTI, N., SEDIVY, J., COOPER, L., VERONDINI, E., MILANESI, L., AND CASTELLANI, G. Reconstructing networks of pathways via significance analysis of their intersections. *BMC Bioinformatics 9*, Suppl 4 (2008), S9.

[42] FREEMAN, L. C. A set of measures of centrality based on betweenness. *Sociometry 40*, 1 (March 1977), 35–41.

[43] FREUDENBERG, J., SIVAGANESAN, S., WAGNER, M., AND MEDVEDOVIC, M. A semi-parametric bayesian model for unsupervised differential co-expression analysis. *BMC Bioinformatics 11*, 1 (2010), 234.

[44] FULLER, T., GHAZALPOUR, A., ATEN, J., DRAKE, T., LUSIS, A., AND HORVATH, S. Weighted gene coexpression network analysis strategies applied to mouse weight. *Mammalian Genome 18* (2007), 463–472.

[45] GEORGE, T., AND MERUGU, S. A scalable collaborative filtering framework based on co-clustering. In *Proceedings of the Fifth IEEE International Conference on Data Mining* (Washington, DC, USA, 2005), ICDM '05, pp. 625–628.

[46] GETZ, G., LEVINE, E., AND DOMANY, E. Coupled two-way clustering analysis of gene microarray data. *Proc. Natl. Acad. Sci. USA 97* (2000), 12079–12084.

[47] GILL, R., DATTA, S., AND DATTA, S. A statistical framework for differential network analysis from microarray data. *BMC Bioinformatics 11*, 1 (2010), 95.

[48] GOLUB, T. R., SLONIM, D. K., TAMAYO, P., HUARD, C., GAASENBEEK, M., MESIROV, J. P., COLLER, H., LOH, M. L., DOWNING, J. R., CALIGIURI, M. A.,

BLOOMFIELD, C. D., AND LANDER, E. S. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science 286*, 5439 (Oct. 1999), 531–537.

[49] GUBICHEV, A., BEDATHUR, S., SEUFERT, S., AND WEIKUM, G. Fast and accurate estimation of shortest paths in large graphs. In *Proceedings of the 19th ACM international conference on Information and knowledge management* (New York, NY, USA, 2010), CIKM '10, pp. 499–508.

[50] GUO, Z., LI, Y., GONG, X., YAO, C., MA, W., WANG, D., LI, Y., ZHU, J., ZHANG, M., YANG, D., AND WANG, J. Edge-based scoring and searching method for identifying condition-responsive proteinprotein interaction sub-network. *Bioinformatics 23*, 16 (2007), 2121–2128.

[51] HATFIELD, G. W., HUNG, S.-P., AND BALDI, P. Differential analysis of dna microarray gene expression data. *Molecular Microbiology 47*, 4 (2003), 871–877.

[52] HAVELIWALA, T. H. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge and Data Engineering 15* (2003), 784–796.

[53] HIPPO, Y., TANIGUCHI, H., TSUTSUMI, S., MACHIDA, N., CHONG, J.-M., FUKAYAMA, M., KODAMA, T., AND ABURATANI, H. Global gene expression analysis of gastric cancer by oligonucleotide microarrays. *Cancer Research 62*, 1 (2002), 233–240.

[54] HO, J., AND CHARLESTON, M. Network modelling of gene regulation. *Biophysical Reviews 3* (2011), 1–13.

[55] HO, J. W., STEFANI, M., DOS REMEDIOS, C. G., AND CHARLESTON, M. A. Differential variability analysis of gene expression and its application to human diseases. *Bioinformatics 24*, 13 (2008), i390–398.

[56] HOFFMANN, R., AND VALENCIA, A. Protein interaction: same network, different hubs. *Trends Genet 19*, 12 (Dec. 2003), 681–683.

[57] HSU, K.-W., BANERJEE, A., AND SRIVASTAVA, J. I/O scalable bregman co-clustering. In *PAKDD'08: Proceedings of the 12th Pacific-Asia conference on Advances in knowledge discovery and data mining* (2008), pp. 896–903.

[58] HU, R., QIU, X., GLAZKO, G., KLEBANOV, L., AND YAKOVLEV, A. Detecting intergene correlation changes in microarray analysis: a new approach to gene selection. *BMC bioinformatics 10*, 1 (2009), 20+.

[59] HUANG, D. W. A. . W., SHERMAN, B. T., AND LEMPICKI, R. A. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature protocols 4*, 1 (Dec. 2009), 44–57.

[60] HUDSON, N. J., REVERTER, A., AND DALRYMPLE, B. P. A differential wiring analysis of expression data correctly identifies the gene containing the causal mutation. *PLoS Comput Biol 5*, 5 (05 2009), e1000382.

[61] HUSSAIN, S. F., AND BISSON, G. Text categorization using word similarities based on higher order co-occurrences. In *SDM* (2010), pp. 1–12.

[62] IDEKER, T., OZIER, O., SCHWIKOWSKI, B., AND SIEGEL, A. F. Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics 18*, suppl 1 (2002), S233–S240.

[63] IDEKER, T., THORSSON, V., SIEGEL, A. F., AND HOOD, L. E. Testing for differentially-expressed genes by maximum-likelihood analysis of microarray data. *Journal of Computational Biology 7*, 6 (2000), 805–817.

[64] IHMELS, J., BERGMANN, S., AND BARKAI, N. Defining transcription modules using large-scale gene expression data. *Bioinformatics 20*, 13 (2004), 1993–2003.

[65] IHMELS, J., BERGMANN, S., BERMAN, J., AND BARKAI, N. Comparative gene expression analysis by a differential clustering approach: Application to the candida albicans transcription program. *PLoS Genet 1*, 3 (09 2005), e39.

[66] IRINA MORDUKHOVICH, PAUL L. REITER, D. M. B. L. F. L. E. M. K. M. O. H. R., AND OLSHAN, A. F. A review of african american-white differences in risk factors for cancer: prostate cancer. *Cancer Causes and Control 2*, 3 (2011), 341–357.

[67] JI, L., AND TAN, K.-L. Mining gene expression data for positive and negative co-regulated gene clusters. *Bioinformatics 20*, 16 (2004), 2711–2718.

[68] JIANG, D., TANG, C., AND ZHANG, A. Cluster analysis for gene expression data: A survey. *IEEE Trans. on Knowl. and Data Eng. 16*, 11 (Nov. 2004).

[69] JOVOV, B., ARAUJO-PEREZ, F., SIGEL, C. S., STRATFORD, J. K., MCCOY, A. N., YEH, J. J., AND KEKU, T. Differential gene expression between african american and european american colorectal cancer patients. *PLoS ONE 7*, 1 (2012), e30168.

[70] K. KAILING, H. K., AND KROGER, P. Density-connected subspace clustering for highdimensional data. In *SDM* (2004), pp. 256–257.

[71] KENNEDY, G. C., MATSUZAKI, H., DONG, S., LIU, W. M., HUANG, J., LIU, G., SU, X., CAO, M., CHEN, W., ZHANG, J., LIU, W., YANG, G., DI, X., RYDER, T., HE, Z., SURTI, U., PHILLIPS, M. S., JACINO, B. M. T., FODOR, S. P., AND JONES,

K. W. Large-scale genotyping of complex DNA. *Nature Biotechnology 21*, 10 (2003), 1233–7.

[72] KERR, M. K., MARTIN, M., AND CHURCHILL, G. A. Analysis of variance for gene expression microarray data. *Journal of computational biology : a journal of computational molecular cell biology 7*, 6 (Dec. 2000), 819–837.

[73] KIM, Y., KIM, T.-K., KIM, Y., YOO, J., YOU, S., CHOI, S., AND HWANG, D. Principal network analysis: Identification of subnetworks representing major dynamics using gene expression data. *Bioinformatics 27*, 3 (2011), 391–398.

[74] KOSTKA, D., AND SPANG, R. Finding disease specific alterations in the co-expression of genes. *Bioinformatics 20*, suppl 1 (2004), i194–i199.

[75] LAI, Y., WU, B., CHEN, L., AND ZHAO, H. A statistical method for identifying differential genegene co-expression patterns. *Bioinformatics 20*, 17 (2004), 3146–3155.

[76] LANGFELDER, P., AND HORVATH, S. Wgcna: an r package for weighted correlation network analysis. *BMC Bioinformatics 9*, 1 (2008), 559.

[77] LANGVILLE, A. N., AND MEYER, C. D. Deeper inside pagerank. *Internet Mathematics 1* (2004), 335–380.

[78] LAPATA, M. Automatic evaluation of information ordering: Kendall's tau. *Computational Linguistics 32*, 4 (2006), 471–484.

[79] LAPPAS, T., TERZI, E., GUNOPULOS, D., AND MANNILA, H. Finding effectors in social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2010), KDD '10, pp. 1059–1068.

[80] LI, J., SIM, K., LIU, G., AND WONG, L. Maximal quasi-bicliques with balanced noise tolerance: Concepts and co-clustering applications. In *Proc. SIAM Int. Conf. on Data Mining SDM'08* (Apr. 2008), pp. 72–83.

[81] LI, K. C., LIU, C. T., SUN, W., YUAN, S., AND YU, T. A system for enhancing genome-wide coexpression dynamics study. *Proc Natl Acad Sci 101*, 44 (2004), 15561–6+.

[82] LIU, B.-H., YU, H., TU, K., LI, C., LI, Y.-X., AND LI, Y.-Y. Dcgl: an r package for identifying differentially coexpressed genes and links from gene expression microarray data. *Bioinformatics 26*, 20 (2010), 2637–2638.

[83] LIU, J., LI, Z., HU, X., AND CHEN, Y. Biclustering of microarray data with mospo based on crowding distance. *BMC Bioinformatics 10*, Suppl 4 (2009), S9.

[84] LIU, J., YANG, J., AND WANG, W. Biclustering in gene expression data by tendency. In *Proceedings of the 2004 IEEE Computational Systems Bioinformatics Conference* (Washington, DC, USA, 2004), CSB '04, pp. 182–193.

[85] LIU, M., LIBERZON, A., KONG, S. W., LAI, W. R., PARK, P. J., KOHANE, I. S., AND KASIF, S. Network-based analysis of affected biological processes in type 2 diabetes models. *PLoS Genet 3*, 6 (2007), e96.

[86] LIU, Q., DINU, I., ADEWALE, A., POTTER, J., AND YASUI, Y. Comparative evaluation of gene-set analysis methods. *BMC Bioinformatics 8*, 1 (2007), 431.

[87] LIU, Z.-P., WANG, Y., ZHANG, X.-S., XIA, W., AND CHEN, L. Detecting and analyzing differentially activated pathways in brain regions of alzheimer's disease patients. *Mol. BioSyst. 7* (2011), 1441–1452.

[88] LNNSTEDT, I., AND SPEED, T. Replicated microarray data. *Statistica Sinica 12* (2001), 31–46.

[89] LUCCHESE, C., ORLANDO, S., AND PEREGO, R. A generative pattern model for mining binary datasets. In *Proceedings of the 2010 ACM Symposium on Applied Computing* (New York, NY, USA, 2010), pp. 1109–1110.

[90] MA, H., SCHADT, E. E., KAPLAN, L. M., AND ZHAO, H. COSINE: Condition-specific sub-network identification using a global optimization method. *Bioinformatics 27*, 9 (2011), 1290–1298.

[91] MACDONALD, T. J., BROWN, K. M., LAFLEUR, B., PETERSON, K., LAWLOR, C., CHEN, Y., PACKER, R. J., COGEN, P., AND STEPHAN, D. A. Expression profiling of medulloblastoma: PDGFRA and the RAS/MAPK pathway as therapeutic targets for metastatic disease. *Nature Genetics 29*, 2 (Oct. 2001), 143–152.

[92] MADEIRA, S. C., AND OLIVEIRA, A. L. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM 1*, 1 (2004), 24–45.

[93] MINGUEZ, P., AND DOPAZO, J. Assessing the biological significance of gene expression signatures and co-expression modules by studying their network properties. *PloS One 6*, 3 (Mar. 2011), e17474+.

[94] MOVAHEDI, S., VAN DE PEER, Y., AND VANDEPOELE, K. Comparative network analysis reveals that tissue specificity and gene function are important factors influencing the mode of expression evolution in arabidopsis and rice. *Plant Physiology 156*, 3 (2011), 1316–1330.

[95] ODIBAT, O., AND REDDY, C. K. A generalized framework for mining arbitrarily positioned overlapping co-clusters. In *Proceedings of the SIAM International Conference on Data Mining (SDM)* (2011), pp. 343–354.

[96] ODIBAT, O., AND REDDY, C. K. Mining differential hubs in homogenous networks. In *Proceedings of the Ninth Workshop on Mining and Learning with Graphs* (2011), MLG '11, ACM.

[97] ODIBAT, O., AND REDDY, C. K. Ranking differential genes in co-expression networks. In *Proceedings of the ACM Conference on Bioinformatics and Computational Biology (BCB)* (2011), ACM, pp. 350–354.

[98] ODIBAT, O., AND REDDY, C. K. Ranking differential hubs in gene co-expression networks. *Journal of Bioinformatics and Computational Biology 10*, 1 (2012).

[99] ODIBAT, O., REDDY, C. K., AND GIROUX, C. N. Differential biclustering for gene expression analysis. In *Proceedings of the ACM Conference on Bioinformatics and Computational Biology (BCB)* (2010), ACM, pp. 275–284.

[100] OKADA, Y., AND INOUE, T. Identification of differentially expressed gene modules between two-class DNA microarray data . *Bioinformation 4*, 4 (2009), 134–137.

[101] OLDHAM, M. C., HORVATH, S., AND GESCHWIND, D. H. Conservation and evolution of gene coexpression networks in human and chimpanzee brains. *Proceedings of the National Academy of Sciences 103*, 47 (2006), 17973–17978.

[102] PENSA, R. G., AND BOULICAUT, J.-F. Constrained co-clustering of gene expression data. In *SDM* (2008), pp. 25–36.

[103] PRELIC, A., BLEULER, S., ZIMMERMANN, P., WILLE, A., BUHLMANN, P., GRUISSEM, W., HENNIG, L., THIELE, L., AND ZITZLER, E. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics 22*, 9 (May 2006), 1122–1129.

[104] PRIETO, C., RIVAS, M., SNCHEZ, J., LPEZ-FIDALGO, J., AND DE LAS RIVAS, J. Algorithm to find gene expression profiles of deregulation and identify families of disease-altered genes. *Bioinformatics 22*, 9 (2006), 1103–1110.

[105] PRITCHARD, C. C., HSU, L., DELROW, J., AND NELSON, P. S. Project normal: Defining normal variance in mouse gene expression. *Proceedings of the National Academy of Sciences 98*, 23 (2001), 13266–13271.

[106] PRULJ, N. Biological network comparison using graphlet degree distribution. *Bioinformatics 23*, 2 (2007), e177–e183.

[107] R RENEE REAMS, DEEPAK AGRAWAL, M. B. D. S. Y. F. T. O. N. K. J. M. H. T. A. S. S., AND SOLIMAN, K. F. Microarray comparison of prostate tumor gene expression in African-American and Caucasian American males: a pilot project study. *Infect Agent Cancer 4*, suppl 1 (2009).

[108] REMONDINI, D., O'CONNELL, B., INTRATOR, N., SEDIVY, J. M., NERETTI, N., CASTELLANI, G. C., AND COOPER, L. N. Targeting c-Myc-activated genes with a correlation method: Detection of global changes in large gene expression network dynamics. *Proc Natl Acad Sci U S A 102*, 19 (2005), 6902–6906.

[109] REVERTER, A., INGHAM, A., LEHNERT, S. A., TAN, S.-H., WANG, Y., RATNAKUMAR, A., AND DALRYMPLE, B. P. Simultaneous identification of differential gene expression and connectivity in inflammation, adipogenesis and cancer. *Bioinformatics 22*, 19 (2006), 2396–2404.

[110] ROSE, A., SATAGOPAN, J., ODDOUX, C., ZHOU, Q., XU, R., OLSHEN, A., YU, J., DASH, A., JEAN-GILLES, J., REUTER, V., GERALD, W., LEE, P., AND OSMAN, I. Copy number and gene expression differences between african american and caucasian american prostate cancer. *Journal of Translational Medicine 8* (2010), 1–9.

[111] RUAN, J., DEAN, A., AND ZHANG, W. A general co-expression network-based approach to gene expression analysis: comparison and applications. *BMC Systems Biology 4*, 1 (2010), 8.

[112] SERIN, A., AND VINGRON, M. DeBi: Discovering differentially expressed biclusters using a frequent itemset approach. *Algorithms for Molecular Biology 6*, 1 (2011), 18.

[113] SHAN, H., AND BANERJEE, A. Residual bayesian co-clustering for matrix approximation. In *Proc. SIAM International Conference on Data Mining* (2010), pp. 223–234.

[114] SHI, X., FAN, W., AND YU, P. S. Efficient semi-supervised spectral co-clustering with constraints. *IEEE International Conference on Data Mining* (2010), 1043–1048.

[115] SILVA, C. L., SILVA, M. F., FACCIOLI, L. H., PIETRO, R. C., CORTEZ, S. A., AND FOSS, N. T. Differential correlation between interleukin patterns in disseminated and chronic human paracoccidioidomycosis. *Clin Exp Immunol 101*, 2 (1995), 314–20.

[116] SMOOT, M. E., ONO, K., RUSCHEINSKI, J., WANG, P.-L., AND IDEKER, T. Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics 27*, 3 (2011), 431–432.

[117] SONG, S., AND BLACK, M. Microarray-based gene set analysis: a comparison of current methods. *BMC Bioinformatics 9*, 1 (2008), 502.

[118] SONG, Y., PAN, S., LIU, S., WEI, F., ZHOU, M. X., AND QIAN, W. Constrained coclustering for textual documents. In *AAAI* (2010).

[119] SOUTHWORTH, L. K., OWEN, A. B., AND KIM, S. K. Aging mice show a decreasing correlation of gene expression within genetic modules. *PLoS Genet 5*, 12 (2009), e1000776.

[120] STEINBACH, M., KARYPIS, G., AND KUMAR, V. A comparison of document clustering techniques. In *KDD-2000 Workshop on Text Mining, August 20* (2000), M. Grobelnik, D. Mladenic, and N. Milic-Frayling, Eds., pp. 109–111.

[121] STEUER, R., KURTHS, J., DAUB, C. O., WEISE, J., AND SELBIG, J. The mutual information: Detecting and evaluating dependencies between variables. *Bioinformatics 18*, suppl 2 (2002), S231–240.

[122] STUART, J. M., SEGAL, E., KOLLER, D., AND KIM, S. K. A Gene-Coexpression Network for Global Discovery of Conserved Genetic Modules. *Science 302*, 5643 (Oct. 2003), 249–255.

[123] SUBRAMANIAN, A., TAMAYO, P., MOOTHA, V. K., MUKHERJEE, S., EBERT, B. L., GILLETTE, M. A., PAULOVICH, A., POMEROY, S. L., GOLUB, T. R., LANDER, E. S., AND MESIROV, J. P. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America 102*, 43 (2005), 15545–15550.

[124] TAN, P.-N., STEINBACH, M., AND KUMAR, V. *Introduction to Data Mining*. Addison-Wesley, 2005.

[125] TATEBE, K., ZEYTUN, A., RIBEIRO, R., HOFFMANN, R., HARROD, K., AND FORST, C. Response network analysis of differential gene expression in human epithelial lung cells during avian influenza infections. *BMC Bioinformatics 11*, 1 (2010), 170.

[126] TAYLOR, I. W., LINDING, R., WARDE-FARLEY, D., LIU, Y., PESQUITA, C., FARIA, D., BULL, S., PAWSON, T., MORRIS, Q., AND WRANA, J. L. Dynamic modularity in protein interaction networks predicts breast cancer outcome. *Nature Biotechnology 27*, 2 (Feb. 2009), 199–204.

[127] TESSON, B., BREITLING, R., AND JANSEN, R. Diffcoex: a simple and sensitive method to find differentially coexpressed gene modules. *BMC Bioinformatics 11*, 1 (2010), 497.

[128] TORKAMANI, A., AND SCHORK, N. J. Identification of rare cancer driver mutations by network reconstruction. *Genome Research 19*, 9 (2009), 1570–1578.

[129] TUSHER, V. G., TIBSHIRANI, R., AND CHU, G. Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences of the United States of America 98*, 9 (April 2001), 5116–5121.

[130] ULTSCH, A., PALLASCH, C., BERGMANN, E., AND CHRISTIANSEN, H. *A Comparison of Algorithms to Find Differentially Expressed Genes in Microarray Data*. 2010, pp. 685–+.

[131] VAN NAS, A., GUHATHAKURTA, D., WANG, S. S., YEHYA, N., HORVATH, S., ZHANG, B., INGRAM-DRAKE, L., CHAUDHURI, G., SCHADT, E. E., DRAKE, T. A., ARNOLD, A. P., AND LUSIS, A. J. Elucidating the role of gonadal hormones in sexually dimorphic gene coexpression networks. *Endocrinology 150*, 3 (2009), 1235–1249.

[132] VOY, B. H., SCHARFF, J. A., PERKINS, A. D., SAXTON, A. M., BORATE, B., CHESLER, E. J., BRANSTETTER, L. K., AND LANGSTON, M. A. Extracting gene networks for low-dose radiation using graph theoretical algorithms. *PLoS Comput Biol 2*, 7 (07 2006), e89.

[133] WALLACE, T. A., PRUEITT, R. L., YI, M., HOWE, T. M., GILLESPIE, J. W., YFANTIS, H. G., STEPHENS, R. M., CAPORASO, N. E., LOFFREDO, C. A., AND AMBS, S. Tumor immunobiological differences in prostate cancer between african-american and european-american men. *Cancer Research 68*, 3 (2008), 927–936.

[134] WALLEY, A. J., JACOBSON, P., FALCHI, M., BOTTOLO, L., ANDERSSON, J. C., PETRETTO, E., BONNEFOND, A., VAILLANT, E., LECOEUR, C., VATIN, V., JERNAS, M., BALDING, D., ETTENI, M., PARK, Y. S., AITMAN, T., RICHARDSON, S., SJOSTROM, L., CARLSSON, L. M. S., AND FROGUEL, P. Differential coexpression analysis of obesity-associated networks in human subcutaneous adipose tissue. *International Journal of Obesity* (2011).

[135] WANG, Y., CONG, G., SONG, G., AND XIE, K. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2010), KDD '10, ACM, pp. 1039–1048.

[136] WATSON, M. CoXpress: differential co-expression in gene expression data. *BMC Bioinformatics 7*, 1 (2006), 509.

[137] WEN, Z., AND LIN, C.-Y. On the quality of inferring interests from social neighbors. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2010), KDD '10, pp. 373–382.

[138] WHITFIELD, M. L., FINLAY, D. R., MURRAY, J. I., TROYANSKAYA, O. G., CHI, J.-T., PERGAMENSCHIKOV, A., MCCALMONT, T. H., BROWN, P. O., BOTSTEIN, D., AND CONNOLLY, M. K. Systemic and cell type-specific gene expression patterns in scleroderma skin. *Proceedings of the National Academy of Sciences 100*, 21 (2003), 12319–12324.

[139] WU, B. Cancer outlier differential gene expression detection. *Biostatistics 8*, 3 (2006), 566–575.

[140] XIAO, Y., FRISINA, R., GORDON, A., KLEBANOV, L., AND YAKOVLEV, A. Multivariate search for differentially expressed gene combinations. *BMC Bioinformatics 5*, 1 (2004), 164.

[141] XU, J., STOLK, J. A., ZHANG, X., SILVA, S. J., HOUGHTON, R. L., MATSUMURA, M., VEDVICK, T. S., LESLIE, K. B., BADARO, R., AND REED, S. G. Identification of differentially expressed genes in human prostate cancer using subtraction and microarray. *Cancer Research 60*, 6 (2000), 1677–1682.

[142] XU, M., KAO, M.-C. C., NUNEZ-IGLESIAS, J., NEVINS, J. R., WEST, M., AND ZHOU, X. J. J. An integrative approach to characterize disease-specific pathways and their coordination: a case study in cancer. *BMC genomics 9 Suppl 1* (2008).

[143] XU, X., LU, Y., TUNG, A. K. H., AND WANG, W. Mining shifting-and-scaling co-regulation patterns on gene expression profiles. In *Proceedings of the 22nd International Conference on Data Engineering* (2006), ICDE '06, p. 89.

[144] XULVI-BRUNET, R., AND LI, H. Co-expression networks: graph properties and topological comparisons. *Bioinformatics 26*, 2 (2010), 205–214.

[145] ZHANG, B., LI, H., RIGGINS, R. B., ZHAN, M., XUAN, J., ZHANG, Z., HOFFMAN, E. P., CLARKE, R., AND WANG, Y. Differential dependency network analysis to identify condition-specific topological changes in biological networks. *Bioinformatics 25*, 4 (2009), 526–532.

[146] ZHANG, B., TIAN, Y., JIN, L., LI, H., SHIH, I.-M., MADHAVAN, S., CLARKE, R., HOFFMAN, E. P., XUAN, J., HILAKIVI-CLARKE, L., AND WANG, Y. DDN: A caBIG analytical tool for differential network analysis. *Bioinformatics* (2011).

[147] ZHANG, H., SONG, X., WANG, H., AND ZHANG, X. MIClique: An algorithm to identify differentially coexpressed disease gene subset from microarray data. *Journal of biomedicine & biotechnology 2009* (2009).

[148] ZHANG, S., CAO, J., KONG, Y. M., AND SCHEUERMANN, R. H. Go-bayes: Gene ontology-based overrepresentation analysis using a bayesian approach. *Bioinformatics 26*, 7 (2010), 905–911.

ABSTRACT

DIFFERENTIAL MODELING FOR CANCER MICROARRAY DATA

by

OMAR ODIBAT

August 2012

Advisor: Chandan K. Reddy

Major: Computer Science

Degree: Doctor of Philosophy

Capturing the changes between two biological phenotypes is a crucial task in understanding the mechanisms of various diseases. Most of the existing computational approaches depend on testing the changes in the expression levels of each single gene individually. In this work, we proposed novel computational approaches to identify the differential genes between two phenotypes. These approaches aim to quantitatively characterize the differences between two phenotypes and can provide better insights and understanding of various diseases. The purpose of this thesis is three-fold. Firstly, we review the state-of-the-art approaches for differential analysis of gene expression data.

Secondly, we propose a novel differential network analysis approach that is composed of two algorithms, namely, *DiffRank* and *DiffSubNet*, to identify differential hubs and differential subnetworks, respectively. In this approach, two datasets are represented as two networks , and then the problem of identifying differential genes is transformed to the problem of comparing two networks to identify the most differential network components. Studying such networks can provide valuable knowledge about the data. The *DiffRank* algorithm ranks the nodes of two networks based on their differential behavior using two novel differential measures: differential connectivity and differential betweenness centrality for each node. These measures are propagated through the network and are optimized to capture the local and global struc-

tural changes between two networks. Then, we integrated the results of this algorithm into the proposed differential subnetwork algorithm which is called *DiffSubNet*. This algorithm aims to identify sets of differentially connected nodes. We demonstrated the effectiveness of these algorithms on synthetic datasets and real-world applications and showed that these algorithms identified meaningful and valuable information compared to some of the baseline methods that can be used for such a task.

Thirdly, we propose a novel differential co-clustering approach to efficiently find arbitrarily positioned difeferntial (or discriminative) co-clusters from large datasets. The goal of this approach is to discover a distinguishing set of gene patterns that are highly correlated in a subset of the samples (subspace co-expressions) in one phenotype but not in the other. This approach is useful when the biological samples are assumed to be heterogenous or have multiple subtypes. To achieve this goal, we propose a novel co-clustering algorithm, **R**anking-based **A**rbitrarily **P**ositioned **O**verlapping **C**o-Clustering (*RAPOCC*), to efficiently extract significant co-clusters. This algorithm optimizes a novel ranking-based objective function to find arbitrarily positioned co-clusters, and it can extract large and overlapping co-clusters containing both positively and negatively correlated genes. Then, we extend this algorithm to discover discriminative co-clusters by incorporating the class information into the co-cluster search process. The novel discriminative co-clustering algorithm is called **Di**scriminative *RAPOCC* (*Di-RAPOCC*), to efficiently extract the discriminative co-clusters from labeled datasets. We also characterize the discriminative co-clusters and propose three novel measures that can be used to evaluate the performance of any discriminative subspace algorithm. We evaluated the proposed algorithms on several synthetic and real gene expression datasets, and our experimental results showed that the proposed algorithms outperformed several existing algorithms available in the literature.

The shift from single gene analysis to the differential gene network analysis and differential co-clustering can play a crucial role in future analysis of gene expression and can help in understanding the mechanism of various diseases.

AUTOBIOGRAPHICAL STATEMENT

OMAR ODIBAT

Dr. Odibat was born in Jarash, Jordan on January, 1 1980. He raised in Soof a small town in the North. His life there framed a great deal of his thinking. Dr. Odibat finished his high school with being among the top in his school. In 2003, Dr. Odibat was awarded a Bachelor of Science degree with honor in Computer Science from the Yarmouk University in Jordan. He was among the top five in his class. He received a Master of Science degree with honor in Computer Science in 2005 from the University of Jordan. In 2006, he married Noor Alaydie, who has been his best friend and companion. They have two daughters, Sarah, who is 4 years old, and Hala, who is 1 year old.

Following four semesters of teaching at the University of Jordan in Jordan, Dr. Odibat then moved to Detroit, Michigan to pursue graduate studies in Computer Science - data mining and bioinformatics. He earned another Master of Science degree in computer science from Wayne State University in 2011. His Ph.D. thesis demonstrates that the shift from single gene analysis to the differential network analysis and differential co-clustering can play a crucial role in future analysis of gene expression and can help in understanding the mechanism of various diseases.