A Stable and Robust Method to Identify Modules of Functionally Coherent Genes

by

Mandeep Kaur Takhar

B.Sc., The University of British Columbia, 2008
B.Sc., The University of British Columbia, 2006

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Computer Science)

The University Of British Columbia
(Vancouver)

October 2014

© Mandeep Kaur Takhar, 2014
Abstract

Complex cellular functions are carried out by the coordinated activity of networks of genes and gene products. In order to understand mechanisms of disease and disease pathogenesis, it is crucial to develop an understanding of these complex interactions. Microarrays provide the potential to explore large scale cellular networks by measuring the expression of thousands of genes simultaneously. The purpose of our project is to develop a stable and robust method that can identify, from such gene expression data, modules of genes that are involved in a common functional role. These modules can be used as a first step in systems scale analyses to extract valuable information from various gene expression studies. Our method constructs modules by identifying genes that are co-expressed across many diseases. We use peripheral blood microarray samples from patients having one of several diseases and cluster the genes in each disease group separately. We then identify genes that cluster together across all disease groups to construct our modules. We first use our method to construct baseline peripheral blood modules relevant to the lung using 5 groups of peripheral blood microarray samples that were collected as controls for separate studies. An enrichment analysis using gene sets from a number of pathway and ontology databases reveals the biological significance of our modules. We utilize our background modules by doing an enrichment analysis on a list of genes that were differentially expressed in a COPD case vs. control study and identify modules that are enriched in that list.

Although a similar approach has been used to identify modules of genes that are coordinately expressed across multiple conditions, we show that our method is an improvement as it is robust to the order in which the different disease datasets are
presented to the algorithm. We also apply our procedure to 3 different datasets including a COPD dataset, a COPD normal dataset and a lung tissue dataset. We then assess the stability of our method by performing a resampling experiment on our module construction procedure and find that our method repeatedly produces modules with high concordance which is measured by Jaccard distance.
Preface

The research in this thesis was carried out under the supervision of Dr. Raymond T. Ng. I was the primary researcher in all work presented.

Part of the data was provided by The PROOF Centre of Excellence and was collected from Chronic Obstructive Pulmonary Disease (COPD) patients enrolled in the Evaluation of COPD Longitudinally to Identify Predictive Surrogate Endpoints (ECLIPSE) Study. Part of the data was collected by Dr. Anne Ellis at Queen’s University. The remaining data was obtained from publicly available sources. Data used in Chapters 4 and 5 was prepared by The PROOF Centre of Excellence.
# Table of Contents

Abstract ................................................................. ii

Preface ................................................................. iv

Table of Contents .................................................... v

List of Tables ........................................................ viii

List of Figures ........................................................ ix

1 Introduction ......................................................... 1
   1.1 Background ..................................................... 1
   1.2 Motivation and contribution .................................. 2
   1.3 Thesis outline ................................................ 2

2 Related Work ....................................................... 4
   2.1 Clustering based methods .................................... 5
       2.1.1 Chassabel et al. method ............................... 5
       2.1.2 Other relevant clustering based methods to derive functional modules ........................................... 6
   2.2 Reverse engineering approaches ............................ 9
       2.2.1 WGCNA method ....................................... 9
       2.2.2 Other relevant reverse engineering methods to derive functional modules .................................... 10
   2.3 Summary .......................................................... 12
# Module Implementation

3.1 K-means clustering
- Selecting the number of clusters
- Selecting the distance function
- Parameters used

3.2 Cluster matching
- Cluster matching algorithm
- Cluster matching and switched labels
- Order independence

3.3 Module extraction

3.4 Evaluation of the module construction procedure
- Jaccard distance and order independence

3.5 Background modules
- Data
- Preprocessing
- Modules

# Module Characteristics

4.1 Stability
- Stability within datasets
- Stability across datasets

4.2 Sensitivity
- Number of samples
- Number of modules
- Module size

# Application of Baseline Modules

5.1 Application to COPD
- Description of COPD
- Description of COPD study
- Using modules for the discovery of transcriptional biomarkers
5.1.4 Differential expression analysis on a module-by-module basis ........................................ 46
5.2 Baseline module functional annotation ................................................................. 57
  5.2.1 Tissue enrichment .................................................................. 57
  5.2.2 Pathway enrichment .................................................................. 58

6 Conclusion ................................................................................................................. 63

Bibliography .................................................................................................................. 65
List of Tables

Table 3.1 Information of samples used ................................. 27
Table 3.2 List of parameters used in the construction of background modules 27
Table 3.3 List of modules created and their respective sizes along with annotation information of probe-sets included in each module ................................. 30

Table 4.1 Split information. This table outlines the number of subsets used in each split (each row represents one split) of the data and the proportion of overlapping sample in each subset as well as the number of repetitions used to sample the data set .................. 34
Table 4.2 P-values resulting from t-tests performed on the pairwise average Jaccard2 distances for each split of the data. ............. 38

Table 5.1 Sample information .................................................. 46
Table 5.2 Baseline modules that were enriched for probe-sets that are differentially expressed in IE vs. NE samples as determined by the LiMMa moderated t-test and an adjusted p-value cut-off of 0.05. Those modules having an adjusted p-value < 0.01 are highlighted in bold ................................. 47
Table 5.3 A list of the 24 modules that were significantly enriched (adjusted pvalue <0.01) in one of the 3 lists of differentially expressed genes. The 3 lists include genes that were found to be differentially expressed by moderated t-tests performed using 30 day IE vs. NE samples, 60 day IE vs. NE samples, and 90 day IE vs. NE samples ................................. 51
List of Figures

Figure 2.1  (a) Module Construction Algorithm Overview [11] and (b) pseudo code [2]  .................................................. 7

Figure 3.1  Matching algorithm applied to diseases d1 and d2 where the genes in each disease, represented by letters A-Z, were partitioned into k=5 clusters each. (a) depicts the d1-optimal results when matching with respect to d1 whereas (b) depicts the d2-optimal solution when matching with respect to d2.  ........  19

Figure 3.2  Cluster matching algorithm with the carry forward parameter implemented. The figure on the left shows a d1-optimal solution (purple) whereas the right side shows a d2-optimal solution (purple). Each solution contains 8 clusters only one of which is not identical in both solutions.  ....................... 20

Figure 3.3  Illustration of iteratively applying the cluster matching algorithm to extract modules from our simulated data set of 4 diseases, 26 genes, and 5 cluster centers.  ....................... 22
Figure 3.4 Evaluation of module concordance using Jaccard distance when varying the carry forward parameter from 0% to 40% (a) indicates the change in Jaccard1 distance, in red, and the change in Jaccard2 distance, in blue. The Jaccard distances are plotted along the y-axis and the mean Jaccard distance across all permutations are indicated in the text. In (b) we’ve illustrated the change in runtime, in minutes, when the carry forward parameter is increased. The time in seconds is indicated in the text.

Figure 3.5 Preprocessing steps used on microarray data before the module construction procedure was applied to construct background modules.

Figure 3.6 Visualization of the separation in our modules by the first principle component of a PCA analysis.

Figure 4.1 Average pairwise Jaccard2 distances between modules constructed using our module construction procedure on multiple data sets created by repeated random sampling of three original data sets (a) Peripheral blood samples from patients having COPD, (b) lung tissue samples from normal subjects, and (c) peripheral blood samples from COPD control subjects.

Figure 4.2 The relationship between input sample size and Jaccard2 distance. Two subsets of the COPD data with a varying number of samples are created by repeated random sampling, with a varying number of repetitions used. The module construction procedure is applied to both subsets and the average Jaccard2 distance is calculated.

Figure 4.3 The relationship between the number of input samples used and the number of modules constructed.

Figure 4.4 Relationship between the size of the universe of probe-sets, number of modules, and input sample size.

Figure 4.5 The distribution of module size. The colors indicate the range of the number of modules constructed.
Figure 5.1  Venn diagram indicating the advantage of performing differential expression analysis on a module-by-module basis rather than a differential expression analysis on the universe of probe-sets that pass the pre-processing steps.

Figure 5.2  (a) Venn diagram showing the 391 probe-sets differentially expressed in either the 30 day IE vs. NE samples or 60 day IE vs NE samples that were found in our modules. Of the 391 probe-sets, 132 were differentially expressed in both lists, 182 were differentially expressed in only the 60 day IE vs. NE samples and 77 were differentially expressed in only the 30 day IE v. NE samples. (b) An illustration of the differences in mean fold-change of genes that were significant in the 30 day IE vs. NE samples compared to the mean fold-change of genes that were significant in the 60 day IE vs. NE samples. Blue circles represent a mean negative fold-change and red circles represent a mean positive fold-change. The number of up-regulated probe-sets and the number of down-regulated probe-sets are indicated by the text in each circle.

Figure 5.3  The first principal component of module genes enriched for IE vs. NE differentially expressed genes was computed. The samples were stratified into 30 day IE samples, 60 day IE samples and NE samples and projected onto the first principal component. The boxplots on the left indicate the separation of these samples by the first principal component and show the deviation of IE samples from NE samples. The 30 day IE samples deviate farther than the 60 day IE samples. The figure on the right illustrates the same information in module M46. Here, the 30 day IE samples deviate from the NE samples less than the 60 day IE samples.
Figure 5.4 Genes that were up regulated and down regulated in IE samples compared to NE samples. The genes that were differentially expressed in the 30 day IE group were compared with genes that were differentially expressed in the 60 day IE group. The venn diagram on the left depicts this information for Module M5 whereas the venn diagram depicts this information for Module M46.

Figure 5.5 Module M5 enrichment in KEGG pathways and GO biological process. A hypergeometric overlap of genes in module M5 and genes in KEGG pathways and GO biological process categories was performed and a p-value cut-off of 0.05 was applied.

Figure 5.6 Median enrichment values of the genes in each module. The different tissues are plotted along the x-axis and are grouped by tissue type. The 53 baseline modules are plotted along the y-axis.

Figure 5.7 KEGG pathways associated with our modules. Only modules that had at least one significant (adjusted p-value <0.01) pathway association are shown.

Figure 5.8 Gene Ontology biological processes associated with our modules. Only modules that had at least one significant (adjusted p-value <0.01) biological process associated with them are shown.
Chapter 1

Introduction

1.1 Background
Complex cellular functions are carried out by the coordinated activity of networks of genes and gene products. In order to understand mechanisms of disease and disease pathogenesis, it is crucial to develop an understanding of these complex interactions in addition to an understanding of individual gene functions [3]. High throughput methods for assaying gene expression levels such as microarray experiments produce mass amounts of data and have become a standard tool used in biomedical research. Since microarrays measure the expression of thousands of genes simultaneously, providing a snapshot of all transcriptional activity taking place in a sample, they provide the potential to explore large scale cellular networks [9]. Microarray experiments have frequently focused on identifying changes in gene expression between case-control study groups. Such analyses have often led to lists of differentially expressed genes that have been used as biomarkers and have contributed to the development of new diagnostic tools [5]. However many of these types of studies encounter analytical challenges as microarray results are prone to include noise which ensues false positive results, and the results often are inconsistent across different laboratories and platforms. Furthermore the biological and functional interpretation of such lists of genes has been difficult and is often the bottleneck of many studies when it comes to characterizing diseases and understanding disease pathogenesis. In recent years as an attempt to mitigate
these problems, there has been a considerable focus on uncovering modules consisting of genes involved in a common functional role as a first step in systems scale analyses [5–7]. Several different computational approaches have been used to construct such modules [8–11]. The approach focused on in this thesis is the co-expression based approach similar to those used by Chaussabel et al. to create transcriptional modules that map the human immune system [11] and by Horvath et al. in Weighted Gene Co-expression Analysis (WGCNA) [42].

1.2 Motivation and contribution

The motivation of our project was to develop a stable and robust method that can identify modules of genes that are involved in a common functional role which can be used as a first step in systems scale analyses and can also be used to extract valuable information in various studies. Our method builds on previous work done by Chaussabel et al. which identifies genes that are co-expressed across many diseases. We improved this method by making modifications to ensure that it is robust to the order in which different diseases are presented to the algorithm and used it to construct baseline modules pertinent to the lung. We also assessed the stability of our method to ensure that we produced stable modules even when using varying samples and different datasets. Furthermore, we aimed to compare our method to a popular reverse engineering approach to construct modules by applying that method to the same samples and datasets used by our method and comparing the results. Lastly, we demonstrated the benefits of using the modules constructed by our procedure for biomarker discovery. As a case-study, we used COPD which is a poorly characterized disease consisting of multiple phenotypes.

1.3 Thesis outline

The rest of this thesis is organized as follows. In Chapter 2 we’ve provided an overview of related work focusing on clustering based methods and reverse engineering approaches. In Chapter 3 we’ve provided a detailed description of the implementation of our module construction procedure as well as an assessment of the quality of our procedure and finally we applied our procedure to lung data to create baseline modules and illustrated those resulting modules. In Chapter 4
we’ve measured the stability and sensitivity of our module construction procedure. We assessed the stability within data sets as well as across different data sets and we assessed the sensitivity to sample size and module size. In Chapter 5 we’ve used our baseline modules to enhance biomarker discovery analysis. Finally our conclusions have been detailed in Chapter 6.
Chapter 2

Related Work

Several groups have worked on inferring global regulatory networks by identifying strongly co-regulated genes. Most approaches to this fall under two broad categories: clustering based approaches and reverse engineering approaches. The networks resulting from these methods have frequently been used post hoc to gain biological insight into experimental results or to advance biological knowledge of the regulatory interactions of specific genes or processes. Fewer groups have worked on the identification of modules of genes to use in a more untargeted fashion, for example as a first step in differential expression analysis where the goal would be to find networks of genes that are conserved across conditions but are up or down regulated in case vs control groups. Even fewer studies have used modules of genes for differential co-expression analysis where the aim is to identify networks of genes where expression is conserved across case vs. controls but network structure is perturbed. Further, we found very few studies that identified modules of genes and used the perturbations of those modules as the basis for building diagnostic tools or classifiers of poorly characterized conditions, which is the goal of the work done in this thesis. The focus of this thesis is on a clustering based approach however the rest of this chapter will outline popular methods of both categories.
2.1 Clustering based methods

Clustering allows one to identify groups of genes that share the same expression pattern across a number of different experimental conditions. Since genes having similar expression patterns have a good chance of being functionally related, clustering algorithms have often been applied to microarray expression data with the aim of determining functional relationships among genes and to help infer regulatory networks. The most common clustering methods applied to gene expression data are hierarchical clustering [11, 12, 31] and k-means or variations of k-means clustering [7, 9, 11, 27, 28].

2.1.1 Chaussabel et al. method

The Chaussabel et al. group designed a strategy for constructing modules by identifying groups of genes that are coordinately expressed across multiple diseases [11]. The reasoning behind this strategy is that the likelihood of many genes having the same expression pattern across many diseases just by chance is low therefore these modules could represent biologically functional units.

In this module construction method, K-means clustering is applied to microarray gene expression data of each disease group separately to cluster the transcripts into 30 clusters. The number of clusters to use was chosen to be K=30 by the elbow-criterion which requires clustering the datasets numerous times while varying the number of clusters, K, each time. The ratio of 'between cluster variance' and 'within cluster variance' is then assessed for each value of K. The issue of selecting the number of clusters for k-means clustering is discussed further in Section 3.1.1.

Module selection from the clustered data sets is done in a few iterative steps. The procedure starts with analyzing cluster membership and selecting the largest set of genes belonging to the same cluster in all diseases. This set is then expanded to include genes belonging to the same cluster in any combination of n-1 diseases, n-2 diseases, and so forth. The reduction in the number of diseases included in cluster membership analysis is implemented to account for the fact that in a given functional unit, some genes may not be expressed in every condition. All genes belonging to the newly created module are then removed from the pool of genes to
select from for the next iteration. This module selection process, which is depicted in Figure 2.1, is repeated again with the next largest set of co-clustering genes and is continued iteratively until all genes have been considered. A functional annotation is assigned to each module using a literature profiling method that associates keyword occurrence in PubMed abstracts with the genes in each module [10].

Chaussabel et al. constructed modules to represent the transcriptional activity of the human immune system by using peripheral blood mononuclear (PBMC) samples from individuals with one of 8 immune-mediated diseases. After three rounds of selection, their method resulted in 28 modules. This modular framework has been used to gain biological insight in studies involving poorly characterized diseases where the understanding of disease pathogenesis is incomplete [4, 7, 27, 28]. These modules have also been used to conduct differential expression analysis at the modular level by averaging the proportion of each module containing differentially expressed genes between 2 groups [9, 11].

2.1.2 Other relevant clustering based methods to derive functional modules

Eisen et al. [12] used a form of hierarchical clustering on both yeast and human gene expression data to separate genes into functional categories. This clustering method is used to assemble all elements into a single tree represented as a dendrogram. The branches of the resulting tree are the desired clusters. A similarity matrix containing similarity scores for all pairs of genes is computed and used as input into their clustering algorithm. The clustering algorithm starts with the most similar pair of genes and creates a node by joining the two genes and averaging their expression profiles. The similarity matrix is then updated by replacing the two joined genes with the new node and this process is repeated until only one element remains. Their method was applied to a single time course of a model of growth response in human cells as well as time course data of the yeast cell cycle. In both types of data they found clusters of genes that share similar expression patterns over multiple conditions. To ensure the biological origin of this pattern, they
Figure 2.1: (a) Module Construction Algorithm Overview [11] and (b) pseudo code [2]
randomized the human growth response data in 3 ways and found no similar clustering patterns. They also found that many groups of genes that were co-expressed and represented diverse patterns of expression across the measured conditions were involved in shared cellular processes.

Segal et al. [32] applied both hierarchical clustering and the expectation-maximization method, which involves a similar procedure to k-means clustering. They identified regulatory modules in yeast stress data using a set of known regulatory genes in addition to gene expression data. Using hierarchical clustering they created an initial set of 50 modules and then identified for each module a regulation program, which is a small set of regulatory genes that control the mRNA expression of the genes in that module. The regulation program is structured as a regression tree and is identified through a search over the space of all possible trees. Each gene is then re-assigned to the module whose regulation program best predicts its behavior. These two steps are implemented using the Expectation-Maximization method and are repeated until convergence. They evaluated the functional coherence of their modules using external data sets which resulted in 46 coherent modules. Since the yeast cell cycle and stress response has been thoroughly studied and well documented, they were able to evaluate the ability of their method to derive regulation from expression.

Another group of researchers implemented a biclustering method [30] to identify co-regulated groups of genes which were then used to infer global regulatory networks using their own network inference procedure [8, 15]. They used a biclustering method where both genes as well as conditions were clustered, to allow some genes to be grouped into multiple clusters as they may be involved in multiple cellular processes and also to account for the chance that some groups of genes may be co-regulated only in subsets of conditions, similar to the rationale behind some of Chaussabel et al.’s implementation. Similar to Segal et al.’s incorporation of a priori knowledge to their clustering method, Bonneau et al. integrated known functional associations and the occurrence or detection of sequence motifs with the biclustering of genes and conditions. While many groups commonly use these types of associations to assess the quality and biological relevance of their
clusters post-hoc, Bonneau et al. treated these associations as priors with appropriately assigned weights (based on prior knowledge and biological relevance). This approach helps reduce the rate of false positives compared to clustering methods alone. Their clustering method was applied to a Halobacterium data set which led to 300 biclusters. These biclusters were used as input to their regulatory network inference procedure [8] which identified a network of 1,431 regulatory influences of varying strengths. This regulatory network inference procedure has more recently been enhanced further by incorporating into it structure priors from multiple data sources [15].

2.2 Reverse engineering approaches

Reverse engineering approaches use graphs to model the structure of co-regulation networks where genes are represented by nodes and the co-regulation relationships between genes are represented by edges. The co-regulation relationships and relative strengths are typically defined by adjacency values. The selection of the adjacency metric is an important step in these methods and not a trivial one. Once the adjacency measure is selected, several different approaches can be used to remove spurious edges and extract important co-regulation modules. An important aspect of these approaches is the identification of hub genes in the resulting modules and networks. Hub genes are the few highly connected nodes that account for most of the connections in the network and link the rest of the less connected nodes to the system. Hub genes are important in regulatory network inference as they likely have a fundamental role and may be drivers of disease due to their vital positions in the networks.

2.2.1 WGCNA method

The weighted gene co-expression network analysis (WGCNA) method implemented by Horvath et al. constructs networks or modules based on co-expression of the most highly varying genes in microarray gene expression data [42]. This method builds on pre-existing network concepts such as scale-free networks and constructs modules consisting of genes that have high topological overlap, that is, genes that are roughly connected to the same group of genes in the network. Hub genes are
central to the topology of this type of network as they link the rest of the less connected nodes to the system and provide robustness to errors and false connections. An important feature of the WGCNA method is the use of a soft threshold. In this network a connection strength, or weight, between two genes is derived from the Pearson correlation and is used to define how a pair of nodes is connected rather than using the typical threshold cut-off and a binary value to represent connected or un-connected nodes. Using this soft-threshold prevents information loss and allows one to maintain the continuous spectrum between complete co-expression and no co-expression. The topological overlap matrix, derived from the adjacency matrix of connection strengths, is then used with average linkage hierarchical clustering to define modules in the network. This module construction process has been used on various data sets for different studies including simulated data, yeast data, and chimpanzee data in addition to human data. After constructing modules, Horvath et al. use the first principal component of a module expression matrix, known as the eigen-gene, to represent a module. This reduction of each module into one expression profile facilitates the use of analysis methods frequently used in patient-based microarray studies, such as differential expression and causal analysis, but at the modular level. The application of WGCNA modules has been illustrated in various studies [13, 18, 29, 40].

2.2.2 Other relevant reverse engineering methods to derive functional modules

Another frequently used reverse engineering approach called ARACNe (Algorithm for the Reconstruction of Accurate Cellular Networks) infers global regulatory networks [26]. This method uses mutual information to identify the gene co-regulation relationships. Mutual information of two genes measures the genes mutual dependence. A mutual information threshold is also applied in this method to eliminate any spurious non-significant edges. After generating the gene co-regulation relationships, ARACNe uses data processing inequality to eliminate indirect relationships between genes. Indirect relationships between two genes are those where the genes are separated by two or more links with other genes. These types of genes may be co-expressed even though there is no direct regulatory interaction between them. This differentiation between direct and indirect relationships between
genes reduces false positive and is a fundamental difference between ARACNe and other co-expression networks. When applied to expression data of human B cells, ARACNe identified a major hub which led to the identification of new target genes controlled by the MYC transcription factor which is known to regulate a network of known target genes [5].

GeneNet is another well-developed method that infers global regulatory networks [31]. GeneNet is a Gaussian graphical model which assumes gene expression values are jointly Gaussian distributed. In this method pairwise gene co-regulation relationships are defined by partial correlation estimates where the correlation of any two genes is conditioned on the remainder of the genes. This method uses improved estimates of the correlation matrix by using regularization methods which also makes this method suitable for situations where the sample size is small relative to the number of variables as in microarray data. As in the ARACNe method, this graphical model is also able to distinguish between direct and indirect interactions. To obtain the final network, the significance of the correlations is tested and edges that represent spurious correlations are differentiated from those that should be included in the resulting network. To accomplish those tasks, GeneNet uses a method called the local fdr network search which calculates the posterior probability for the presence/absence of an edge, and takes account of the multiplicity in the simultaneous testing of edges. This method when applied to gene expression data from a breast cancer study containing 49 samples and expression measurements of 7129 genes also revealed several highly connected hub genes involved in key processes such as the control of tumor growth [31].

Another group of researches implemented a different method to infer global regulatory networks called GENIE3 (GEne Network Inference with Ensemble of Trees) [20]. In their method, the problem of finding estimates of co-regulation is converted to a feature selection problem: for any given target gene they define the identification of its regulatory genes as determining the subset of genes whose expression is predictive of the expression of the target gene. This method therefore is able to predict directed edges to represent co-regulation relationships. In this method tree based ensemble methods (Random Forests and Extra Trees) are used to solve the
feature selection problem. The tree-based methods are able to deal with combinatorial regulations and non-linear relationships which is an advantage of this method over other reverse engineering approaches. Tree based methods provide a variable importance measure, which is a ranking of features by their importance in predicting the output variable. Once this ranking is computed for all genes a threshold can be applied, similar to many other reverse engineering approaches, to remove the edges that are thought to be spurious. One of the datasets they tested GENIE3 on was the E.Coli dataset for which prior, experimentally validated, information on which genes are transcription factors was available. Their method performed very well on this dataset however, when prior information was unavailable, the performance of their method dropped.

2.3 Summary

Many of the reverse engineering approaches aim to identify all global transcriptional interactions or some transcriptional interactions with high confidence. These methods have been used in a targeted way to identify transcriptional regulators of specific genes and to further the understanding of known regulatory processes. Because the WGCNA method focuses on identifying modules of tightly connected genes rather than a global regulatory network, it can be used in a more untargeted fashion in differential co-expression analysis which is more suitable for the analysis of poorly characterized conditions. To look for perturbations of networks or modules, a highly robust method of constructing modules is required. Chaussabel et al.’s clustering approach to module construction provides this robustness by looking at clustering patterns of samples collected under many different experimental conditions. Another advantage of this framework is that it allows flexibility in the clustering method and similarity measure as well as flexibility in the datasets used. Since the datasets are clustered separately, it provides potential to extend its use to more heterogeneous data sets from multiple sources and platforms.
In Chapter 2 we introduced the approach used by Chaussabel et al. to construct modules that were used to create a map of the human immune system. The basis of their method was the identification of genes that co-cluster across multiple conditions. The method we’ve used to implement our modules also uses that notion as our method builds on the work done by Chaussabel et al. In this chapter we’ve provide detailed descriptions of the steps involved in our implementation.

3.1 K-means clustering

Clustering methods are applied to data to partition all data points into distinct clusters where the points in any given cluster are closer, or more similar, to each other than to points belonging to different clusters. This similarity of data points is a similarity in terms of some variables of interest and is typically determined by some distance function [17]. Numerous methods can be used to perform clustering analysis but in our module construction algorithm we’ve used k-means clustering to cluster each of our datasets. The k-means clustering algorithm is a top down technique as it randomly divides all data points into a predetermined number of clusters. It then iteratively reassigns each data point to the cluster that has a center that is closest to that point. The following two key steps are repeated until
convergence in the algorithm [17]:

1. For each data point, identify the closest cluster center (measured by a distance function).

2. Replace each cluster center by the average of all data points that are closest to it.

In our module construction procedure, k-means was applied separately to the dataset of each disease group. As mentioned above this algorithm uses a predetermined number of clusters therefore, we can be sure that each disease dataset is being partitioned into the same number of clusters.

3.1.1 Selecting the number of clusters

Selecting the number of clusters for k-means is a subjective matter and the appropriate number to use can be ambiguous. Chaussabel et al. chose 30 clusters for the k-means step of their module construction process. This number was decided upon using the frequently utilized elbow criterion where the proportion of variance explained or the within sum of squares for each clustering is a function of the number of clusters. Ideally the proportion of variance explained increases with the number of clusters until a certain point where the curve levels off as the increase in number of clusters no longer proves a large increase in proportion of variance explained. That point indicates the optimal number of clusters. Chaussabel et al. selected the largest number of optimal clusters indicated by this method for any of their datasets. We’ve maintained the use of 30 clusters for the k-means step of our module construction process as 30 clusters explained approximately 97% of the variance in our datasets and an increase beyond 30 provided very little increase in the proportion of variance explained.

3.1.2 Selecting the distance function

Selecting a dissimilarity measure, or distance function, is fundamental to all clustering methods. The k-means algorithm is intended for use with data containing quantitative variables and with Euclidean distance as the chosen dissimilarity measure [17]. Therefore we have selected Euclidean distance as our distance function.
with our variables of interest being quantitative measurements of the expression of each gene on the microarray. Euclidean distance is defined as:

$$d(x_i, x_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{ij'})^2 = ||x_i - x_{i'}||$$

(3.1)

Using Euclidean distance on genes assumes that every gene is "equi-distant" from any other gene. This is likely an over simplifying assumption and creates a bias towards round clusters. Our module construction procedure can easily be adapted to use a number of dissimilarity measures to partition our data into k clusters by making minor changes such as partitioning around medoids rather than means.

### 3.1.3 Parameters used

As mentioned above, k-means clustering partitions data points into k clusters so that the sum of squares from data points to the assigned cluster’s center is minimized. Since the initial k centers are arbitrarily chosen, the solution will likely not be the minimal sum of squares of all possible partitions [16]. Instead, a local minimum is returned where movement of a data point from one cluster to another won’t reduce the within cluster sum of squares any further. One option to help arrive at a more stable solution is to attempt multiple initial configurations and report the one delivering the best solution. In our module construction procedure we’ve used 10 random initial configurations which can also be adjusted by varying one parameter. Although this provides a minimum over several partitions, it still does not guarantee a global minimum.

Because the k-means algorithm iterates over two fundamental steps until convergence with the upper bound on its runtime being exponential in the number of data points [37], we’ve limited the number of iterations to 50 in order to achieve a reasonable execution runtime.
3.2 Cluster matching

In our module construction procedure, each disease condition was separately partitioned into 30 clusters using k-means. Because a module should be composed of genes that cluster together in multiple conditions, it was essential to identify corresponding clusters across all conditions. Cluster matching is a non-trivial issue that becomes problematic due to two properties that are inherent to the clustering process. The first issue arises due to the stochastic properties of the k-means algorithm. Since the k-means algorithm starts with a random initial configuration of centers and there is no guarantee of a solution that is the global minimum, replicate cluster analyses can produce different solutions. In our case of clustering across different biological conditions, there were also biological factors contributing to differences in cluster membership in the different groups. The second is a rather trivial problem caused by the method of label assignment to the clusters. Labels are arbitrarily assigned during clustering therefore even with replicate clustering analyses on the same data, clusters with identical contents may take on different labels.

3.2.1 Cluster matching algorithm

To resolve the issue of matching clusters across different clustering analyses, we considered it as a graph matching problem where the sets of clusters generated by each clustering analysis were the disjoint sets to match. We used a relaxed variation of the weighted stable marriage problem to solve it. The stable marriage problem (SMP) is a well-known problem of matching elements in one set to elements in another disjoint set. The elements in each set have associated with them a list of strictly ordered preferences for all elements in the other set and SMP provides a matching such that no two elements, which are not matched to each other, both prefer each other over their current matches [14]. In the weighted stable marriage problem, each element provides a score or a weight rather than a ranking for elements in the other set. There exist more general versions of SMP that allow for partial ordering of preferences which is suitable for practical applications where an alternative pairing is not possible for an element of a set [21]. We used this generalization in our matching algorithm as clusters in one set will only match with all
clusters in another set in the worst case.

One further adjustment we made to the generalized SMP was to allow for a many-to-one pairing so multiple elements of one set may be paired with the same element of the second set. This was desired because we were interested only in the genes that clustered together in multiple conditions and these co-clustered genes may be a subset of larger clusters in each of the conditions, or some conditions may have merged or split clusters relative to another condition.

As an example, which we’ve illustrated in Figure 3.1, we used a simulated data set of 26 probe-sets represented by letters A-Z and 4 disease groups which have been arbitrarily relabeled as d1-d4. The probe-sets in each disease group were partitioned into k=5 clusters by the k-means algorithm. When matching two sets of clusters (d1-d2 or d3-d4), we followed these steps:

1. Consider 2 sets of clusters (d1 and d2)

2. Initialize an adjacency matrix using clusters of d1 as the rows and clusters of d2 as columns

3. Compute the length of the intersection between each row and each column and assign that as a weight in the matrix. For example C1 of d1 intersects with C7, C8 and C10 of d2 and the respective lengths of those intersections are 2, 3, and 1 as indicated by the numbers on the edges between clusters in Figure 3.1(a)

4. For each row, select the column with the largest weight, breaking ties arbitrarily, thereby maximizing the overlap between clusters in d1 and clusters in d2

The algorithm provides a solution for this example that is d1-optimal which is illustrated in Figure 3.1(a). Changing the order in which we select pairings, by switching columns and rows or selecting the cluster pairs column-wise for example, will render a different solution that is d2-optimal as illustrated in Figure 3.1(b).
rows and therefore the largest weight is being selected in terms of $d_2$. In doing so, the resulting solution differs by 4 clusters: $\{\text{DOY}\}$ and $\{\text{HPS}\}$ from Figure 3.1(a) are missing whereas $\{\text{K}\}$ and $\{\text{WZ}\}$ are now present. In Figure 3.1(a) clusters $C_1$, $C_2$ and $C_3$ of $d_1$ all intersect with a subset of cluster $C_8$ of $d_2$ each having a weight of 3. Since that is the largest weight in all three cases, when selecting weights in terms of $d_1$, all three intersections will be chosen resulting in the clusters $\{\text{CMV}\}$, $\{\text{DOY}\}$, $\{\text{HPS}\}$. However when selecting weights in terms of $d_2$, as is the case in Figure 3.1(b), only one of those intersections will be chosen as $C_1$, $C_2$ and $C_3$ are a tie for $C_8$ and ties are broken arbitrarily. Therefore the cluster $\{\text{CMV}\}$ is a part of the resulting solution but clusters $\{\text{DOY}\}$ and $\{\text{HPS}\}$ are not.

This characteristic of multiple solutions is common and acceptable for many graph matching problems. However in our case, we required the matching algorithm to render one unique solution because this algorithm was applied iteratively to extract modules, which we have described further in Section 3.3, and only one set of possible modules should be constructed for a given set of disease conditions regardless of the order in which disease groups are presented to the algorithm.

3.2.2 Cluster matching and switched labels

In order to resolve the issue of switched labels, we used a hash to maintain a universe of clusters. Multiple conditions were clustered sequentially and the hash of clusters was updated with any novel clusters that arose from each clustering analysis. If the clustering algorithm identified clusters that already existed in the hash, the labels of those clusters were replaced to match the labels stored in the hash. This way clusters across all conditions used a uniform labelling convention so the problem of switched labels was eliminated.

3.2.3 Order independence

To ensure that constructed modules varied minimally when different permutations of disease order were input to the clustering procedure, we made one more adjustment to the algorithm. Rather than selecting the pairing with the largest weight, we carried forward a list of multiple pairings for each cluster. To prevent the al-
Figure 3.1: Matching algorithm applied to diseases d1 and d2 where the genes in each disease, represented by letters A-Z, were partitioned into k=5 clusters each. (a) depicts the d1-optimal results when matching with respect to d1 whereas (b) depicts the d2-optimal solution when matching with respect to d2.

From exhausting considerable time and space, we introduced a threshold parameter that was used as a cutoff to specify the maximum number of pairings to carry forward for each cluster. We set that parameter to be equal to 40% of the number of centers specified in k-means. Further details on selecting this value are provided in Section 3.4. In our implementation k = 30 therefore the parameter = 30(0.4) = 12. In our simulated example k = 5 so the parameter was then 5(0.4) = 2. In Figure 3.2 we’ve illustrated the result of the algorithm with the carry forward parameter implemented using two diseases in different orders.
Although implementing the carry forward parameter does not guarantee a unique solution, it does provide a notable increase in the concordance among solutions. Without the carry forward parameter, there was only a 60% overlap in solutions as illustrated in Figure 3.1 with only 3 out of 5 clusters being identical. However, once the carry forward parameter was implemented, 88% of clusters were identical as illustrated in Figure 3.2 with 7 out of 8 identical clusters. We evaluated the order independence of our entire module construction procedure further and have provided more details in Section 3.4.

3.3 Module extraction

To construct modules after clustering each disease group, the algorithm described in sections 3.2.1 and 3.2.3 was applied iteratively. The module extraction process is illustrated using our simulated example in Figure 3.3. We implemented a size cut-off of 30 genes to avoid obtaining small modules that consist of only 1 or 2 genes. Using our simulated example, we followed these steps:

1. Apply k-means using 5 cluster centers to each disease group (note that each disease has arbitrarily been relabeled d1 through d4)
2. Add all clusters created in step 1 to the empty cluster universe

3. Initialize two adjacency matrices using d1 and d3 clusters as rows and d2 and d4 clusters as columns

4. Compute the lengths of the intersection between each row and column and assign that as a weight in each matrix

5. In each matrix, select 2 \([5*0.4=2]\) columns with the largest weights for each row to carry forward and add the intersection to sets \(d1 \cap d2\) and \(d3 \cap d4\)

6. Remove the genes selected in step 4 from the cluster universe

7. Repeat steps 2-4 using the newly created \(d1 \cap d2\) and \(d3 \cap d4\) clusters as rows and columns and continue this procedure until only one set results

8. Select all clusters from the resulting set with size greater than the specified cut-off and add to the module set

9. Create a module that indicates no assignment and assign to it any probe-sets that were not assigned to a module
Figure 3.3: Illustration of iteratively applying the cluster matching algorithm to extract modules from our simulated data set of 4 diseases, 26 genes, and 5 cluster centers.
3.4 Evaluation of the module construction procedure

We assessed the quality of our module construction process by considering a few different parameters. We observed the results after varying the number of probe-sets and cluster centers, and the number and order of diseases, as well as the carry forward parameter. We also used these results to select the value to use for the carry forward parameter in our implementation.

3.4.1 Jaccard distance and order independence

To evaluate module concordance when presenting the algorithm with different permutations of the diseases, we used Jaccard1 and Jaccard2 distances. Jaccard1 distance is defined as:

$$ J1(A, B) = \frac{|A \cap B|}{|A \cup B|} $$

(3.2)

where A and B are different clusters. Jaccard2 distance is defined as:

$$ J2(A, B) = \frac{|A \cap B|}{\min\{|A|, |B|\}} $$

(3.3)

Jaccard1 and Jaccard2 indicate a different measure of similarity when comparing clusters of unequal sizes and when some clusters are subsets of other clusters. While Jaccard1 provides an unbiased measure of similarity between 2 clusters by treating both clusters equally regardless of their sizes, Jaccard2 is more sensitive to the presence of subsets. For example if cluster B is a subset of cluster A, Jaccard1 would indicate a less than complete similarity even though all genes in cluster B are also in cluster A. Jaccard2 however would show complete similarity as the overlap of the 2 clusters is scaled by the length of the smaller cluster.

When using N number of diseases we ran the module construction procedure using each possible permutation of disease order, thereby running the algorithm N! times. However, in the interest of time when N > 6 we used only 1000 randomly selected permutations of all possible permutations. In Figure 3.4 (a) we’ve plotted the average Jaccard1 distance (in red) and average Jaccard2 distance (in blue) when
the carry forward parameter was varied between 0%, 10%, 20%, 30%, and 40% of cluster centers. We used a simulated example of 3328 probe-sets, 7 diseases, and 30 cluster centers. Each permutation of inputs produces a different set of modules as described in Section 3.2.1. The degree of similarity of the modules resulting from each permutation is dependent on the carry forward parameter as described in Section 3.2.3. When the carry forward parameter is 0% of cluster centers, the average Jaccard distances should be very low which is the result we obtained as indicated in Figure 3.1(a). As the carry forward parameter increases, the resulting solutions of all permutations should begin to converge which would be indicated by higher Jaccard distances. The Jaccard2 distances were higher than the Jaccard1 distances in every case. This result was not surprising as it’s likely in our data that a smaller cluster in one permutation may be a complete subset of a larger cluster in a different permutation because some conditions might split or merge clusters relative to other conditions as we described in Section 3.2.1. We can see from the graph that module concordance does indeed increase with the carry forward parameter. In Figure 3.4(b) we’ve also indicated how the runtime of the module construction procedure varied with each value of the carry forward parameter. Since using 40% as the carry forward parameter constructed modules with high concordance (J1=0.86 and J2=0.95) when using different permutations of diseases and also ran in only several minutes (495.62 seconds), we used that value in the implementation of our module construction procedure.

### 3.5 Background modules

We used our module construction procedure to create background modules using normal control samples from various studies involving the lung. These background modules were constructed to provide a snapshot of the complex networks of co-expressed genes that exist in the body. Structuring the data in this way allows us to focus the analysis of studies involving the lung on these sets of functionally coherent genes rather than individually and independently testing thousands of genes measured by microarrays. The data used and preprocessing steps involved are described in this section and presented in Figure 3.5.
Figure 3.4: Evaluation of module concordance using Jaccard distance when varying the carry forward parameter from 0% to 40% (a) indicates the change in Jaccard1 distance, in red, and the change in Jaccard2 distance, in blue. The Jaccard distances are plotted along the y-axis and the mean Jaccard distance across all permutations are indicated in the text. In (b) we’ve illustrated the change in runtime, in minutes, when the carry forward parameter is increased. The time in seconds is indicated in the text.
3.5.1 Data

To construct background modules we used whole blood samples of five control
groups collected for separate studies. Three of the datasets used included samples
that were collected from COPD patients enrolled in the ECLIPSE study [38]. This
study was a three year longitudinal study which aimed to identify parameters that
predict the progression of disease in different COPD phenotypes. One of addi-
tional two datasets used included samples that were collected by Dr. Anne Ellis
at Queens University. These samples were used in a study done to identify t-cell
differentiation expression patterns associated with pollen exposure in individuals
with allergic rhinitis. The final dataset used included samples that were collected
for a study conducted to identify molecular patterns in peripheral blood of asth-
matic individuals that could differentiate between two response phenotypes after
an allergen inhalation challenge [33].

The COPD control samples used were smoking and non-smoking controls: 29 cur-
rent smokers (CS), 28 former smokers (FS), 29 never smokers (NS). The asthma
control samples used were blood draws done before the inhalation challenges: 14
pre-methacholine inhalation challenge (AR) and 14 pre-allergen inhalation chal-
lenge (Asthma). All samples were processed on the Affymetrix Human Gene ST
platform. COPD samples were processed on Gene ST 1.0 platform whereas the
Asthma samples were processed on the Gene ST 1.1 platform. Sample informa-
tion in summarized in Table 3.1.

3.5.2 Preprocessing

The robust multi-array average (RMA) technique was applied to the microarrays
for background correction and normalization. The Factor Analysis for Robust Mi-
croarray Summarization (FARMS) technique was used for summarization [19]. A
filtering method called I/NI calls was also applied to eliminate non-informative
probe-sets [36]. Since the ECLIPSE samples were processed on a different version
of the Affymetrix Human Gene ST array than the Asthma samples, we were com-
pelled to perform the background correction, normalization, summarization and
pre-filtering steps on the ECLIPSE samples separately from the Asthma samples.
<table>
<thead>
<tr>
<th>Study</th>
<th>Group (Condition)</th>
<th>Number of samples</th>
<th>Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECLIPSE</td>
<td>Current Smokers (CS)</td>
<td>29</td>
<td>Affymetrix Human Gene ST 1.1</td>
</tr>
<tr>
<td>ECLIPSE</td>
<td>Former Smokers (FS)</td>
<td>28</td>
<td>Affymetrix Human Gene ST 1.1</td>
</tr>
<tr>
<td>ECLIPSE</td>
<td>Never Smokers (NS)</td>
<td>29</td>
<td>Affymetrix Human Gene ST 1.1</td>
</tr>
<tr>
<td>Allergic</td>
<td>Pre-methacholine</td>
<td>14</td>
<td>Affymetrix Human Gene ST 1.0</td>
</tr>
<tr>
<td>Rhinitis</td>
<td>Challenge (AR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asthma</td>
<td>Pre-allergen Challenge</td>
<td>14</td>
<td>Affymetrix Human Gene ST 1.0</td>
</tr>
<tr>
<td></td>
<td>(Asthma)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 3.1: Information of samples used**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of cluster centers</td>
<td>30</td>
</tr>
<tr>
<td>Number of random starts</td>
<td>10</td>
</tr>
<tr>
<td>Carry forward</td>
<td>0.4 (40% of cluster centers)</td>
</tr>
<tr>
<td>Minimum module size cut-off</td>
<td>30</td>
</tr>
</tbody>
</table>

**Table 3.2: List of parameters used in the construction of background modules**

After those steps, we were left with 12,561 probe-sets in the ECLIPSE samples and 6,348 probe-sets in the Asthma samples. Taking the intersection of the probe-sets in both groups left us with 5788 probe-sets. This data set of 5 sample groups and 5,788 probe-sets were batch corrected using the ComBat Bioconductor package [22]. The preprocessing steps have been illustrated in Figure 3.5.

### 3.5.3 Modules

We applied our module construction algorithm to the 5 datasets after completing the preprocessing steps. The parameters used during the procedure are listed in Table 3.2. The algorithm produced 53 modules of varying sizes, which are listed in Table 3.3, where the smallest module contained the minimum of 30 probe-sets and the largest contained 155 probe-sets.
Figure 3.5: Preprocessing steps used on microarray data before the module construction procedure was applied to construct background modules

<table>
<thead>
<tr>
<th>Module</th>
<th>No. of unique probe-sets</th>
<th>No. of unique genes</th>
<th>No. of unannotated probe-sets</th>
<th>No. of probe-sets with multiple mappings</th>
<th>Largest probe-set to gene mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>155</td>
<td>108</td>
<td>80</td>
<td>21</td>
<td>8</td>
</tr>
<tr>
<td>M2</td>
<td>135</td>
<td>111</td>
<td>43</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>M3</td>
<td>123</td>
<td>103</td>
<td>27</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>M4</td>
<td>119</td>
<td>93</td>
<td>29</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Module</th>
<th>No. of unique probe-sets</th>
<th>No. unique genes</th>
<th>No. of unannotated probe-sets</th>
<th>No. of probe-sets with multiple mappings</th>
<th>Largest probe-set to gene mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>M5</td>
<td>116</td>
<td>77</td>
<td>51</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>M6</td>
<td>107</td>
<td>51</td>
<td>70</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>M7</td>
<td>98</td>
<td>93</td>
<td>37</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>M8</td>
<td>96</td>
<td>50</td>
<td>63</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>M9</td>
<td>96</td>
<td>82</td>
<td>43</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>M10</td>
<td>94</td>
<td>80</td>
<td>24</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>M11</td>
<td>85</td>
<td>75</td>
<td>28</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>M12</td>
<td>82</td>
<td>35</td>
<td>50</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>M13</td>
<td>82</td>
<td>79</td>
<td>18</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>M14</td>
<td>78</td>
<td>73</td>
<td>9</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>M15</td>
<td>78</td>
<td>52</td>
<td>34</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>M16</td>
<td>74</td>
<td>44</td>
<td>37</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>M17</td>
<td>72</td>
<td>72</td>
<td>17</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>M18</td>
<td>63</td>
<td>58</td>
<td>9</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>M19</td>
<td>62</td>
<td>75</td>
<td>9</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>M20</td>
<td>62</td>
<td>57</td>
<td>9</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>M21</td>
<td>62</td>
<td>55</td>
<td>11</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>M22</td>
<td>58</td>
<td>48</td>
<td>16</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>M23</td>
<td>55</td>
<td>51</td>
<td>7</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>M24</td>
<td>54</td>
<td>35</td>
<td>31</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>M25</td>
<td>54</td>
<td>57</td>
<td>5</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>M26</td>
<td>53</td>
<td>22</td>
<td>36</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>M27</td>
<td>52</td>
<td>47</td>
<td>11</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>M28</td>
<td>52</td>
<td>56</td>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>M29</td>
<td>52</td>
<td>48</td>
<td>11</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>M30</td>
<td>51</td>
<td>21</td>
<td>36</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>M31</td>
<td>51</td>
<td>41</td>
<td>18</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>M32</td>
<td>50</td>
<td>60</td>
<td>3</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>M33</td>
<td>50</td>
<td>44</td>
<td>12</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>
Table 3.3: List of modules created and their respective sizes along with annotation information of probe-sets included in each module

<table>
<thead>
<tr>
<th>Module</th>
<th>No. of unique probe-sets</th>
<th>No. unique genes</th>
<th>No. of unannotated probe-sets</th>
<th>No. of probe-sets with multiple mappings</th>
<th>Largest probe-set to gene mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>M34</td>
<td>50</td>
<td>44</td>
<td>12</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>M35</td>
<td>48</td>
<td>25</td>
<td>30</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>M36</td>
<td>45</td>
<td>47</td>
<td>17</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>M37</td>
<td>45</td>
<td>35</td>
<td>13</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>M38</td>
<td>43</td>
<td>24</td>
<td>20</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>M39</td>
<td>39</td>
<td>35</td>
<td>14</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>M40</td>
<td>38</td>
<td>32</td>
<td>15</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>M41</td>
<td>38</td>
<td>38</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>M42</td>
<td>37</td>
<td>19</td>
<td>34</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>M43</td>
<td>37</td>
<td>33</td>
<td>12</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>M44</td>
<td>36</td>
<td>12</td>
<td>28</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>M45</td>
<td>35</td>
<td>36</td>
<td>9</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>M46</td>
<td>34</td>
<td>27</td>
<td>18</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>M47</td>
<td>34</td>
<td>7</td>
<td>29</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>M48</td>
<td>33</td>
<td>27</td>
<td>8</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>M49</td>
<td>33</td>
<td>25</td>
<td>11</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>M50</td>
<td>33</td>
<td>8</td>
<td>29</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>M51</td>
<td>32</td>
<td>26</td>
<td>7</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>M52</td>
<td>31</td>
<td>29</td>
<td>5</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>M53</td>
<td>30</td>
<td>10</td>
<td>25</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

We performed a principle component analysis (PCA) on the resulting modules to visualize their separation and illustrated the result in Figure 3.6. We’ve plotted the first two principle components and colored each data point according to its corresponding module. The first principle component explained 97.9% of the vari-
Figure 3.6: Visualization of the separation in our modules by the first principal component of a PCA analysis.
Chapter 4

Module Characteristics

4.1 Stability

Our module construction procedure uncovers modules based on the natural biological structure of the data. Since the transcriptional organization in biological systems is conserved and reproducible, a sound and reliable module construction procedure should be able to repeatedly uncover modules that reflect this underlying biology.

In Chapter 3 we used our module construction procedure to construct background modules using normal control samples from various studies involving the lung. To construct our modules, we used samples from multiple studies involving various conditions pertinent to the lung to incorporate the biological variability naturally present across human cellular systems. These background modules are meant to be a universal set of modules that provide a snap shot of the complex networks of co-expressed genes that exist in the body regardless of conditions present. These modules can then be used in numerous studies pertaining to conditions involving the lung. In order to accomplish these tasks, it is particularly essential that our module construction procedure is sound and reliable, repeatedly producing stable modules.

To measure this aspect in our module construction procedure, we used an objective
measure, the Jaccard2 index (equation 3.3), along with permutation testing and re-sampling techniques to show that our procedure uncovers biologically consistent modules even when using varied inputs and different data sets.

4.1.1 Stability within datasets

In our analysis of the robustness of our procedure, we used sampling techniques to create multiple data sets from the original data and applied our procedure to each subset of the data. We then evaluated module similarity across the resulting sets of modules.

The data used was a COPD (described in Section 5.1) data set containing 238 peripheral blood microarray samples from patients having COPD. Although COPD is a condition with varying sub-phenotypes and varying severity of disease (discussed in more detail in Chapter 5), for this task we considered COPD samples as homogeneous and did not stratify the samples by their sub-phenotype.

Splitting of the data was carried out seven times with an increasing number of subsets created each time, starting with 2 subsets and increasing up to 10 subsets of the original data. When creating 2 subsets of data the procedure used was to sample 119 samples from the original 238 to obtain the first subset and use the remaining 119 samples as the second subset. This random sampling was repeated ten times to account for variability. This procedure returned two distinct subsets with no overlapping samples, however when we increased the number of subsets created we used the same sampling procedure with one modification: we also increased the proportion of samples that overlap across the subsets of data. For example, when creating 3 subsets of data, we sampled 79 samples three times to obtain 3 distinct subsets. We then merged two of those subsets using all three combinations of two, giving us three subsets of data with 50% of the samples overlapping across subsets. The number of subsets used and the proportion of overlap in each split of the data have been summarized in Table 4.1.
Table 4.1: Split information. This table outlines the number of subsets used in each split (each row represents one split) of the data and the proportion of overlapping sample in each subset as well as the number of repetitions used to sample the data set.

<table>
<thead>
<tr>
<th>No. of subsets</th>
<th>Proportion of overlap</th>
<th>No. of repetitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>67</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>75</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>80</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>86</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>89</td>
<td>10</td>
</tr>
</tbody>
</table>

Our module construction procedure requires that we have multiple disease groups. In order to emulate that in the COPD data set, we further stratified each subset of data into 3 random groups which were to represent 3 disease groups. This stratification was repeated 5 times for each subset to account for variability.

In each of the seven splits of data, we applied our module construction procedure on each subset and evaluated the concordance among the resulting sets of modules in a pairwise way. The concordance of two sets of modules, set A and set B, was evaluated by finding for each module in set A, the module in set B with the maximum Jaccard distance. The maximum Jaccard distance which indicates the best matches is defined as:

$$\max J^2(A, B) = \forall a \in A, \forall b \in B \max \left( \frac{|a \cap b|}{\min\{|a|, |b|\}} \right)$$

(4.1)

We then computed the average of those Jaccard distances across all modules in set A. Since the maximum Jaccard distance between a pair of modules is not symmetric i.e. $\max J^2(A,B)$ does not always equal $\max J^2(B,A)$, as explained
in Chapter 3, the comparison of two sets of modules was done in a bi-directional way. The average of maximum Jaccard2(A,B) and maximum Jaccard2(B,A) was taken to be the average Jaccard2 distance for set A and set B and that was used as our measure of concordance:

\[
\text{concordance}(A, B) = \frac{1}{|A| + |B|} \sum \left[ \{\text{max}J^2(A, B)\}, \{\text{max}J^2(B, A)\} \right] \quad (4.2)
\]

This process was repeated for all pairwise comparisons in one split of the data. When we have k subsets of data, we generate k sets of modules which gives us \((k*(k-1))/2\) possible pairwise comparisons. For example in our first split we generated 2 sets of modules which gave us 1 pairwise comparison, and in our last split we generated 10 subsets of data which gave us \((10*9)/2 = 45\) pairwise comparisons. The boxplots in Figure 4.1(a) illustrate the result of this analysis where the average maximum Jaccard2 distance from every pairwise comparison is plotted as one point in each split of the data. We can see that when two distinct subsets of data are used, there is slightly over 60% mean concordance in the resulting modules with a small range of variation. This indicates that our module construction procedure can produce modules that reflect the underlying biological consistency in the data. As the data in each subset became more similar, in each subsequent split of the data, the concordance of the resulting modules also increased as expected. These results indicate the robustness of our procedure as the modules were highly similar even with slightly varied input data.

In order to measure the significance of the observed Jaccard2 values, we performed a permutation test procedure where we randomly permuted module assignment before each pairwise comparison. This permutation was repeated 10 times for each pairwise comparison. The results are indicated by the lower line and set of error bars in Figure 4.1. We can see that the permuted labels only reached an average Jaccard2 distance of about 0.10 with very small variance which confirms that our modules are not a random set of genes and are driven by the innate structure in the data.

35
4.1.2 Stability across datasets

We were also interested in evaluating the consistency of our procedure across different data sets. We used two data sets for this purpose. One of those data sets included transcriptional profiles from 76 normal lung tissue samples which were downloaded from Gene Expression Omnibus public repository [23, 41]. Using this data we performed an analysis replicate to the one described in section 4.1.1 and have displayed the result in Figure 4.1(b). The second data set used included 87 peripheral blood samples from COPD control patients. Using this data we again performed a replicate analysis to that described in section 4.1.1 and illustrated the result in Figure 4.1(c). Both of these applications again showed high concordance, measured by the average Jaccard2 values, and a small range of variation in the resulting modules when doing pairwise comparisons. The average Jaccard2 distance in the tissue data set was 0.537-0.704 and in the COPD controls data set it was 0.587-0.739. The results of the permutation analyses were also similar to that seen in the COPD data set as in both new cases we reached an average Jaccard2 distance of only 0.10. This result implies the reliability of our procedure in uncovering underlying biological structure as it produced consistent modules using data from different tissues.

To compare the results across all three data sets we performed a t-test on the average Jaccard2 distances for each split of the data. The results of this test are given in Table 4.2. One notable observation is that those modules generated using the GEO tissue normal samples as input as well as those generated using peripheral blood normal samples as input did not give a higher concordance than modules constructed using the COPD data set. This is contradictory to what was expected as when using only normal controls the samples are expected to be more homogeneous and therefore are likely to produce modules with higher concordance when using different subsets of the data. The outcome that we saw however may be attributable to the sizes of the datasets used, which we discuss further in Section 4.2.
Figure 4.1: Average pairwise Jaccard2 distances between modules constructed using our module construction procedure on multiple data sets created by repeated random sampling of three original data sets (a) Peripheral blood samples from patients having COPD, (b) lung tissue samples from normal subjects, and (c) peripheral blood samples from COPD control subjects.
<table>
<thead>
<tr>
<th>Overlap</th>
<th>COPD vs. GEO</th>
<th>COPD vs. Normals</th>
<th>GEO vs. Normals</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>7.16e-33</td>
<td>6.53e-41</td>
<td>1.28e-53</td>
</tr>
<tr>
<td>50%</td>
<td>1.37e-41</td>
<td>1.096e-84</td>
<td>9.94e-113</td>
</tr>
<tr>
<td>67%</td>
<td>3.92e-08</td>
<td>2.36e-104</td>
<td>9.35e-179</td>
</tr>
<tr>
<td>75%</td>
<td>0.00262027</td>
<td>2.68e-132</td>
<td>2.93e-204</td>
</tr>
<tr>
<td>80%</td>
<td>1.62e-30</td>
<td>7.08e-108</td>
<td>5.43e-249</td>
</tr>
<tr>
<td>86%</td>
<td>4.31e-109</td>
<td>6.93e-90</td>
<td>1.24e-171</td>
</tr>
<tr>
<td>89%</td>
<td>3.77e-220</td>
<td>6.38e-58</td>
<td>9.008e-99</td>
</tr>
</tbody>
</table>

Table 4.2: P-values resulting from t-tests performed on the pairwise average Jaccard2 distances for each split of the data.

4.2 Sensitivity

4.2.1 Number of samples

In the previous sections we showed the result of our module construction procedure using three different data sets and compared the results across data sets. One factor that may contribute to the quality of the results is the size of the input data. The first data set, the COPD data set, was comprised of peripheral blood samples from patients that were classified with different phenotypes of COPD and displayed varying degrees of symptoms. This data set contained 238 samples. The second data set, the GEO data set, included tissue samples from the lungs of 78 normal control samples. This set is considered to be relatively homogeneous compared to the COPD samples since they are controls and they come from tissue as opposed to peripheral blood. The third data set, the COPD controls, included 89 peripheral blood samples from subjects that did not have a lung condition and were used as controls in the ECLIPSE study. We performed an analysis to assess the effect of sample size on module concordance and we’ve illustrated the result in Figure 4.2. We again used the resampling technique to create two subsets of data from the original COPD data set. This random sampling was repeated multiple times to account for variability. This entire repeated random sampling experiment was run 8...
times, and an increasingly larger number of samples were used in each experiment starting from 20 samples and increasing to all 238 samples. When fewer samples were used, a larger number of repetitions were used for random sampling. For each experiment the module construction procedure was applied to both subsets of data in each run and the bi-directional Jaccard2 average was taken between both sets of modules (described previously in Section 4.1.1). Figure 4.2 clearly shows the dependence of module concordance, measured by average maximum Jaccard2 distance, on sample size as the Jaccard2 distance increases from 0.335 when using 20 samples to 0.618 when using 238 samples.

One noteworthy point in the result is the average Jaccard2 distance when 80 samples are used. That sample size is more similar to the lung tissue and COPD control data sets. When comparing this Jaccard2 distance to the 0% overlap columns (two distinct subsets) in Figure 4.1 (b) and (c), we see that the COPD data set has a Jaccard2 distance of 0.481 which is lower than both the lung tissue data (0.537) and COPD normal data (0.587). This is an intuitive result as the lung tissue data set and COPD normal data set consists of samples that are more homogeneous relative to the COPD data set. When the input samples are more similar, the modules constructed are also expected to be more similar.

4.2.2 Number of modules

Since the module construction procedure has no parameter to control the number of modules constructed, unless a replicate data set is used, the number of modules resulting from multiple applications of the procedure will vary. We performed an analysis to assess the relationship between the number of samples in the input data and the number of modules constructed. The same dataset from the previous section was used where two subsets of the original data were repeatedly sampled using a varying number of samples. The result is depicted in Figure 4.3 which shows that a fewer number of samples will produce a larger number of modules. However the number of modules constructed is less consistent when using fewer samples as indicated by the larger range of variation when using only 10 samples to construct modules compared to using 119 samples.
Figure 4.2: The relationship between input sample size and Jaccard2 distance. Two subsets of the COPD data with a varying number of samples are created by repeated random sampling, with a varying number of repetitions used. The module construction procedure is applied to both subsets and the average Jaccard2 distance is calculated.

We also examined the relationship between the size of the universe of probe-sets (union of probe-sets across all modules) and the number of modules constructed. The result is depicted in Figure 4.4 where figures (a) through (c) show the number of modules vs. the size of the universe in each of our 3 data sets. We see a fairly consistent universe size with the number of modules varying between 50 and 120. Figure 4.4(d) includes another layer of information by varying the number of samples used to construct modules and we see that a larger number of samples results in a larger universe of probe-sets distributed over a smaller number of modules.

4.2.3 Module size

Another factor that may contribute to the quality of the constructed modules is the module size distribution i.e. the number of probe-sets in each module. Using the
Figure 4.3: The relationship between the number of input samples used and the number of modules constructed.

Figure 4.4: Relationship between the size of the universe of probe-sets, number of modules, and input sample size.
same dataset we used in Section 4.1.1 (two subsets of the original data, repeatedly sampled for a varying number of samples) we observed the distribution of module sizes. In Figure 4.5 we have depicted this distribution across the varying number of input samples (10 samples - 119 samples). The colors indicate the range of the number of modules created. The sizes of the modules vary between 30 probe-sets and approximately 570 probe-sets. The number of modules created varies between 60 and 200 modules.

**Figure 4.5:** The distribution of module size. The colors indicate the range of the number of modules constructed.
Chapter 5

Application of Baseline Modules

5.1 Application to COPD

In previous chapters we introduced our module construction procedure and illustrated its application in the construction of baseline modules pertinent to the lung. In this chapter we’ve illustrated how the use of those baseline modules can help to enhance analysis methods and we use COPD as a case study.

5.1.1 Description of COPD

COPD is a disease containing multiple phenotypes for which pathogenesis is not completely understood. It is a common preventable and treatable disease characterized by the progressive limitation of airflow associated with an exaggerated chronic inflammatory response in the lungs which is caused by the inhalation of toxins in genetically susceptible individuals. Toxins that contribute to the inflammatory response include noxious particles and gases such as air pollution, infection, and cigarette smoking. Cigarette smoking is the best-studied risk factor for COPD but non-smokers also develop chronic airflow limitation and not all people with the same smoking history will develop COPD which may likely be due to genetic differences. The lung inflammation caused by toxins is a normal response which seems to be modified in patients with COPD [39]. The inflammatory processes cause destruction of lung tissue which leads to further structural changes that disrupt normal repair and defense mechanisms and reduce the ability of the airways
to stay open during expiration.

COPD is a major global health problem which is projected to soon rank fifth in terms of disease burden and third in terms of mortality world-wide [39]. COPD is responsible for 11,000 deaths and even more hospital admissions per year in Canada. There is a significant economic burden associated with COPD as there is a direct relationship between the cost of care and COPD severity. COPD related costs in Canada alone are near $4 billion per year and are estimated to more than double in the next 15 years [1].

One of the primary contributors to the projected increases in economic and disease burden is hospital admission rates caused by acute exacerbation of COPD. Exacerbation is characterized by a worsening of a patient’s respiratory symptoms beyond day to day variations and leads to a change in medication. Symptoms include dyspnea, cough, and sputum production. Exacerbations are brought on by several factors such as bacterial or viral respiratory tract infections and air pollution; however the cause of about one-third of severe exacerbations is unidentified. Exacerbations are critical events in the course of COPD for many reasons: lengthy recovery time, acceleration of the rate of decline in lung function, association with significant mortality, high socioeconomic costs, and negative effect on a patient’s quality of life. Diagnosis of exacerbation relies only on the clinical presentation of a patient conveying an acute change of symptoms beyond normal day to day variation. Beta-2 agonists and systemic corticosteroids are used to treat exacerbations, with the goal of treatment being minimization of the impact of the current exacerbation and prevention of subsequent exacerbations. Exacerbation rates vary greatly from patient to patient and not all COPD patients exacerbate. Patients having 2 or more exacerbation events per year are classified as frequent exacerbators (FE) and the severity of an exacerbation can be mild, moderate, or severe, determined by the extent of treatment required. Despite the fact that not all patients exacerbate, all patients receive the same treatment of corticosteroids and beta-2-agonists because there is no way to risk-stratify patients and the single best predictor of exacerbations is a history of exacerbations [39]. This is not only a costly approach but it puts many COPD patients at unnecessary risk of the adverse effects of corticosteroids.
5.1.2 Description of COPD study

The availability of a simple clinical test such as a blood test for the prognosis of exacerbations would be extremely beneficial as it would allow patients to receive treatment tailored for their specific case which would help safeguard against the administration of drugs to patients who don’t require them. The PROOF Centre’s COPD Biomarker Program aims to develop biomarkers to identify patients at high risk of exacerbations for this purpose. The goal here is to identify blood biomarkers of “imminent exacerbation” (IE) which is defined as an exacerbation occurring shortly after blood collection (within 30 days, 60 days or 90 days). The untargeted biomarker discovery method includes an analytic pipeline consisting of many steps starting with all probe-sets measured on a microarray and narrowing down to a panel of genes. These steps include quality assessment, pre-processing, pre-filtering, univariate ranking, univariate selection, classifier generation, combinatorial biomarker generation, and biomarker evaluation. In Section 5.1.3 we’ve described how our baseline modules can be used to help enhance this biomarker discovery strategy by adding a targeted feature to the pipeline.

5.1.3 Using modules for the discovery of transcriptional biomarkers

We performed differential expression analysis between imminent exacerbators and never exacerbators (NE). Blood samples from patients enrolled in the ECLIPSE study (described previously in Chapter 3) were used. This was a three year longitudinal study with the most well-phenotyped COPD cohort in the world. This cohort included 2,180 patients, 343 smoking control subjects and 233 non-smoking control subjects. For our analysis we used a subset of 110 patients which included 24 IE samples and 86 NE samples. All patients were between the ages of 49 and 75 and included 37 females and 73 males. Fifty five patients in this dataset were smokers and the remaining 54 patients were former smokers. Sample information is outlined in Table 5.1).

Sample pre-processing steps were the same as those used for constructing our baseline modules (Section 3.5.2). All samples were processed on the Affymetrix Human Gene ST 1.0 platform and the RMA technique was applied to the microarrays.
for background correction and normalization. The FARMS technique was used for summarization and I/NI calls filtering method was also applied to eliminate non-informative probe-sets. After these pre-processing steps we were left with 12,381 probe-sets.

### 5.1.4 Differential expression analysis on a module-by-module basis

We first identified probe-sets that were differentially expressed between IE and NE samples using only those IE samples for which the exacerbation occurred within 60 days. We used LiMMa moderated t-test from the Bioconductor package [34] to identify differentially expressed probe-sets. This is analogous to the classical t-test except that empirical Bayes methods are used to borrow information between genes to help with inference about each individual gene [34]. We applied a p-value cut-off and selected all probe-sets with a p-value < 0.05 which returned 1,649 probe-sets. We were interested in determining which of our baseline modules are enriched for these differentially expressed genes so we performed a hypergeometric overlap between the genes in each of our modules and the list of 1,649 differentially expressed genes. To control the false discovery rate we applied the Benjamini and Hochberg method for p-value adjustment. The results are presented in Table 5.2 which lists the modules that were enriched for the differentially expressed probe-sets with an adjusted p-value of < 0.01.
## Table 5.2: Baseline modules that were enriched for probe-sets that are differentially expressed in IE vs. NE samples as determined by the LiMMa moderated t-test and an adjusted p-value cut-off of 0.05. Those modules having an adjusted p-value $<$ 0.01 are highlighted in bold.
We were also interested in observing the effect of performing a differential expression analysis on a module by module basis rather than the entire universe of 12,831 probe-sets. We used the same differential expression analysis and applied it to each of the 53 modules. We took the union of probe-sets that were found to be differentially expressed across all modules and compared this to the union of probe-sets in our modules that were enriched for the list of 1,649 differentially expressed probe-sets (union of "No. DE probe-sets in Module" column in Table 5.2). The Venn diagram in Figure 5.1(a) indicates the difference in the results. Performing the differential expression on all probe-sets and finding their enrichment in our modules gave us 402 probe-sets among our modules that were differentially expressed. Performing differential expression analysis on a module by module basis identified 470 probe-sets that were differentially expressed across all modules. These 470 probe-sets included an additional 74 probe-sets that could not be detected by the former analysis that considers all probe-sets simultaneously. This suggests the benefit in performing differential expression analysis on a module by module basis. When performing differential expression on the complete list of probe-sets a masking effect likely occurs since information is borrowed across the ensemble of genes. Performing the analysis on a module by module basis has an effect similar to an increase in resolution as information is borrowed across genes that are functionally related.

LiMMa performs an adjustment to control the false discovery rate to correct for multiple comparisons as well, using the Benjamini and Hochberg procedure, and provides an associated adjusted p-value. Figure 5.1(b) shows the results of applying an adjusted p-value cut-off of 35%. This cut-off of 35% is much larger than would typically be used in a differential expression analysis however a large cut-off was required in this case in order to obtain a sufficient number of probe-sets when differential expression was performed on the non-modular set of probe-sets. Results of this analysis indicate a reduction in the false discovery rates when performing differential expression on a module-by-module basis.
Figure 5.1: Venn diagram indicating the advantage of performing differential expression analysis on a module-by-module basis rather than a differential expression analysis on the universe of probe-sets that pass the pre-processing steps.

**Differential expression in time-course (30d vs. 60d)**

We were also very interested in observing whether or not there are differences in gene expression of IE patients who exacerbate within 30 days (30 day IE), those who exacerbate within 60 days (60 day IE), and those who exacerbate within 90 days (90 day IE). We used LiMMa again to identify differentially expressed genes between 30 day IE and NE samples. We again used a p-value cutoff of 0.05 and this returned 1,274 probe-sets. We performed the same analysis using only the 90 day IE samples and obtained a list of 1,232 probe-sets. For each of these lists we performed hypergeometric overlaps to identify which of our baseline modules are enriched for these lists. We also applied the Benjamini-Hochberg procedure to correct for the false discovery rate and applied an adjusted p-value cut-off of 0.01. There were 24 modules that were significantly enriched in at least one of the 3 lists of differentially expressed genes and the results are listed in Table 5.3.

<table>
<thead>
<tr>
<th>Module</th>
<th>Cohort</th>
<th>No. probe-sets in DE list</th>
<th>No. probe-sets in module</th>
<th>Overlap</th>
<th>P-value</th>
<th>Adjusted p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>60 day</td>
<td>1649</td>
<td>155</td>
<td>27</td>
<td>2.81E-06</td>
<td>1.86E-05</td>
</tr>
<tr>
<td></td>
<td>90 day</td>
<td>1232</td>
<td>155</td>
<td>16</td>
<td>0.002678</td>
<td>0.007886</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Module</th>
<th>Cohort</th>
<th>No. probe-sets in DE list</th>
<th>No. probe-sets in module</th>
<th>Overlap</th>
<th>P-value</th>
<th>Adjusted p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>M2</td>
<td>30 day</td>
<td>1274</td>
<td>135</td>
<td>18</td>
<td>6.64E-06</td>
<td>0.000176</td>
</tr>
<tr>
<td></td>
<td>60 day</td>
<td>1649</td>
<td>135</td>
<td>24</td>
<td>1.44E-07</td>
<td>1.27E-06</td>
</tr>
<tr>
<td></td>
<td>90 day</td>
<td>1232</td>
<td>135</td>
<td>17</td>
<td>1.56E-05</td>
<td>0.000103</td>
</tr>
<tr>
<td>M5</td>
<td>60 day</td>
<td>1649</td>
<td>116</td>
<td>13</td>
<td>0.004196</td>
<td>0.010065</td>
</tr>
<tr>
<td>M6</td>
<td>30 day</td>
<td>1274</td>
<td>107</td>
<td>11</td>
<td>0.002906</td>
<td>0.011</td>
</tr>
<tr>
<td></td>
<td>60 day</td>
<td>1649</td>
<td>107</td>
<td>15</td>
<td>0.000442</td>
<td>0.001399</td>
</tr>
<tr>
<td></td>
<td>90 day</td>
<td>1232</td>
<td>107</td>
<td>9</td>
<td>0.016983</td>
<td>0.031038</td>
</tr>
<tr>
<td>M7</td>
<td>30 day</td>
<td>1274</td>
<td>98</td>
<td>12</td>
<td>0.001476</td>
<td>0.006981</td>
</tr>
<tr>
<td></td>
<td>60 day</td>
<td>1649</td>
<td>98</td>
<td>22</td>
<td>1.01E-07</td>
<td>1.08E-06</td>
</tr>
<tr>
<td></td>
<td>90 day</td>
<td>1232</td>
<td>98</td>
<td>17</td>
<td>1.52E-06</td>
<td>1.61E-05</td>
</tr>
<tr>
<td>M9</td>
<td>30 day</td>
<td>1274</td>
<td>96</td>
<td>8</td>
<td>0.04886</td>
<td>0.080924</td>
</tr>
<tr>
<td></td>
<td>60 day</td>
<td>1649</td>
<td>96</td>
<td>16</td>
<td>0.000135</td>
<td>0.00055</td>
</tr>
<tr>
<td></td>
<td>90 day</td>
<td>1232</td>
<td>96</td>
<td>12</td>
<td>0.000703</td>
<td>0.00266</td>
</tr>
<tr>
<td>M10</td>
<td>60 day</td>
<td>1649</td>
<td>94</td>
<td>14</td>
<td>0.00023</td>
<td>0.000872</td>
</tr>
<tr>
<td></td>
<td>90 day</td>
<td>1232</td>
<td>94</td>
<td>14</td>
<td>8.49E-06</td>
<td>6.42E-05</td>
</tr>
<tr>
<td>M12</td>
<td>30 day</td>
<td>1274</td>
<td>82</td>
<td>14</td>
<td>6.31E-07</td>
<td>3.35E-05</td>
</tr>
<tr>
<td></td>
<td>60 day</td>
<td>1649</td>
<td>82</td>
<td>11</td>
<td>0.000866</td>
<td>0.002699</td>
</tr>
<tr>
<td>M13</td>
<td>30 day</td>
<td>1274</td>
<td>82</td>
<td>13</td>
<td>1.72E-05</td>
<td>0.000228</td>
</tr>
<tr>
<td></td>
<td>60 day</td>
<td>1649</td>
<td>82</td>
<td>19</td>
<td>4.10E-08</td>
<td>7.25E-07</td>
</tr>
<tr>
<td></td>
<td>90 day</td>
<td>1232</td>
<td>82</td>
<td>12</td>
<td>5.35E-05</td>
<td>0.000315</td>
</tr>
<tr>
<td>M14</td>
<td>30 day</td>
<td>1274</td>
<td>78</td>
<td>11</td>
<td>5.77E-05</td>
<td>0.00051</td>
</tr>
<tr>
<td></td>
<td>60 day</td>
<td>1649</td>
<td>78</td>
<td>19</td>
<td>1.82E-09</td>
<td>4.81E-08</td>
</tr>
<tr>
<td></td>
<td>90 day</td>
<td>1232</td>
<td>78</td>
<td>15</td>
<td>3.91E-08</td>
<td>6.91E-07</td>
</tr>
<tr>
<td>M18</td>
<td>30 day</td>
<td>1274</td>
<td>63</td>
<td>10</td>
<td>3.83E-05</td>
<td>0.000406</td>
</tr>
<tr>
<td></td>
<td>60 day</td>
<td>1649</td>
<td>63</td>
<td>12</td>
<td>1.80E-05</td>
<td>9.54E-05</td>
</tr>
<tr>
<td></td>
<td>90 day</td>
<td>1232</td>
<td>63</td>
<td>7</td>
<td>0.002945</td>
<td>0.008214</td>
</tr>
<tr>
<td>M21</td>
<td>30 day</td>
<td>1274</td>
<td>62</td>
<td>8</td>
<td>0.000795</td>
<td>0.006016</td>
</tr>
<tr>
<td></td>
<td>60 day</td>
<td>1649</td>
<td>62</td>
<td>11</td>
<td>7.33E-05</td>
<td>0.000353</td>
</tr>
<tr>
<td>M25</td>
<td>60 day</td>
<td>1649</td>
<td>54</td>
<td>8</td>
<td>0.002969</td>
<td>0.007493</td>
</tr>
<tr>
<td>M28</td>
<td>30 day</td>
<td>1274</td>
<td>52</td>
<td>10</td>
<td>1.28E-05</td>
<td>0.000225</td>
</tr>
<tr>
<td></td>
<td>60 day</td>
<td>1649</td>
<td>52</td>
<td>12</td>
<td>4.92E-06</td>
<td>2.90E-05</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Module</th>
<th>Cohort</th>
<th>No. probe-sets in DE list</th>
<th>No. probe-sets in module</th>
<th>Overlap</th>
<th>P-value</th>
<th>Adjusted p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>90 day</td>
<td>1232</td>
<td>52</td>
<td>14</td>
<td>2.65E-09</td>
<td>7.02E-08</td>
</tr>
<tr>
<td>M29</td>
<td>60 day</td>
<td>1649</td>
<td>52</td>
<td>14</td>
<td>9.83E-08</td>
<td>1.08E-06</td>
</tr>
<tr>
<td></td>
<td>90 day</td>
<td>1232</td>
<td>52</td>
<td>11</td>
<td>1.15E-06</td>
<td>1.53E-05</td>
</tr>
<tr>
<td>M31</td>
<td>30 day</td>
<td>1274</td>
<td>51</td>
<td>7</td>
<td>0.001575</td>
<td>0.006981</td>
</tr>
<tr>
<td></td>
<td>60 day</td>
<td>1649</td>
<td>51</td>
<td>23</td>
<td>1.11E-16</td>
<td>5.88E-15</td>
</tr>
<tr>
<td></td>
<td>90 day</td>
<td>1232</td>
<td>51</td>
<td>18</td>
<td>1.31E-13</td>
<td>6.96E-12</td>
</tr>
<tr>
<td>M35</td>
<td>30 day</td>
<td>1274</td>
<td>48</td>
<td>7</td>
<td>0.000978</td>
<td>0.006477</td>
</tr>
<tr>
<td></td>
<td>60 day</td>
<td>1649</td>
<td>48</td>
<td>8</td>
<td>0.00124</td>
<td>0.003652</td>
</tr>
<tr>
<td></td>
<td>90 day</td>
<td>1232</td>
<td>48</td>
<td>7</td>
<td>0.000787</td>
<td>0.002781</td>
</tr>
<tr>
<td>M36</td>
<td>60 day</td>
<td>1649</td>
<td>45</td>
<td>8</td>
<td>0.006826</td>
<td>0.013914</td>
</tr>
<tr>
<td></td>
<td>90 day</td>
<td>1232</td>
<td>45</td>
<td>8</td>
<td>0.000973</td>
<td>0.003223</td>
</tr>
<tr>
<td>M37</td>
<td>60 day</td>
<td>1649</td>
<td>45</td>
<td>9</td>
<td>8.33E-05</td>
<td>0.000368</td>
</tr>
<tr>
<td></td>
<td>90 day</td>
<td>1232</td>
<td>45</td>
<td>9</td>
<td>6.89E-06</td>
<td>6.09E-05</td>
</tr>
<tr>
<td>M46</td>
<td>30 day</td>
<td>1274</td>
<td>34</td>
<td>6</td>
<td>0.00131</td>
<td>0.006981</td>
</tr>
<tr>
<td>M49</td>
<td>60 day</td>
<td>1649</td>
<td>33</td>
<td>6</td>
<td>0.001429</td>
<td>0.003987</td>
</tr>
</tbody>
</table>

Table 5.3: A list of the 24 modules that were significantly enriched (adjusted \(p\)-value < 0.01) in one of the 3 lists of differentially expressed genes. The 3 lists include genes that were found to be differentially expressed by moderated t-tests performed using 30 day IE vs. NE samples, 60 day IE vs. NE samples, and 90 day IE vs. NE samples.

We focused on the 30 day IE and 60 day IE samples for the purposes of biomarker discovery and looked at the union of probe-sets from our modules that overlap with the two lists of differentially expressed genes. The results are presented in Figure 5.2(a) which illustrates that there were 391 probe-sets in our modules that were differentially expressed in one of the two lists. Of the list of 1,649 probe-sets differentially expressed in 60 day IE and NE samples, 314 were in our modules. Out of the list of 1,274 probe-sets differentially expressed in 30 day IE and NE samples, 209 were in our modules. There was an overlap of 132 probe-sets in our modules that were differentially expressed in both the 30 day IE and 60 day IE samples.
samples compared to NE samples.

Figure 5.2(b) depicts for each module, the differences in mean fold change of the probe-sets that were differentially expressed in the 30 day IE samples vs. the 60 day IE samples. The numbers of genes that are up or down regulated are indicated as well. This figure reveals very interesting results for example, module M5 had 5 up-regulated probe-sets and 5 down-regulated probe-sets in the 30 day IE samples however in the 60 day IE samples, 10 probe-sets were up-regulated and only 3 were down-regulated. Module M46 also shows an interesting result as most differentially expressed probe-sets were up-regulated in the 30 day IE samples (5 up-regulated and 1 down-regulated) but the reverse signal was seen in the 60 day IE samples as most probe-sets were down-regulated (1 up-regulated and 4 down-regulated). We computed the first principal component of these modules using the significant probe-sets in each module and the 30 day IE samples, 60 day IE samples, and NE samples. In Figure 5.3 we’ve illustrated the separation of the samples in each group by the first principal component. In both modules M5 and M46 the signal of the exacerbators deviated from that of the non-exacerbators however in module M5 the 30 day IE samples deviated more than the 60 day IE samples but in M46 the 30 day IE samples deviated less than the 60 day IE samples. This is an optimistic result as it suggests that at the time of blood draw, the mechanism of exacerbation is already underway and we may be able to use different modules to advise how soon an exacerbation might occur which could help guide personalize treatment.

In Figure 5.4 we’ve indicated for both modules the genes that were up regulated (red) and down regulated (blue) in the 30 day IE samples and the 60 day IE samples. Fourteen out of 19 probe-sets had associated gene annotations in module M5 with the remaining 5 probe-sets being unannotated. In module M46, 4 out of 10 probe-sets had associated gene annotations with the remaining 6 being unannotated. In both modules, the mean fold change of significant probe-sets changed direction between 30 day IE samples and 60 day IE samples as seen in Figure 5.2(b) however, the direction of gene expression in the individual genes didn’t change between these samples but rather there were additional genes differentially expressed.
Figure 5.2: (a) Venn diagram showing the 391 probe-sets differentially expressed in either the 30 day IE vs. NE samples or 60 day IE vs NE samples that were found in our modules. Of the 391 probe-sets, 132 were differentially expressed in both lists, 182 were differentially expressed in only the 60 day IE vs. NE samples and 77 were differentially expressed in only the 30 day IE vs. NE samples. (b) An illustration of the differences in mean fold-change of genes that were significant in the 30 day IE vs. NE samples compared to the mean fold-change of genes that were significant in the 60 day IE vs. NE samples. Blue circles represent a mean negative fold-change and red circles represent a mean positive fold-change. The number of up-regulated probe-sets and the number of down-regulated probe-sets are indicated by the text in each circle.
Figure 5.3: The first principal component of module genes enriched for IE vs. NE differentially expressed genes was computed. The samples were stratified into 30 day IE samples, 60 day IE samples and NE samples and projected onto the first principal component. The boxplots on the left indicate the separation of these samples by the first principal component and show the deviation of IE samples from NE samples. The 30 day IE samples deviate farther than the 60 day IE samples. The figure on the right illustrates the same information in module M46. Here, the 30 day IE samples deviate from the NE samples less than the 60 day IE samples.

Functional interpretation of module M5

We performed a literature search to associate possible functions to the differentially expressed genes. Many of the genes in module M5 have a similar functional role or belong to the same family of genes which is a set of similar genes that originated from one original gene and have similar biochemical roles. Four of the six genes differentially expressed in the 30 day IE samples are involved in immune system functioning. Three genes, TNFRSF1A, IGF2R, PPP6C, are part of the CD family which is a set of genes that encode for proteins involved in immune system func-
Figure 5.4: Genes that were up regulated and down regulated in IE samples compared to NE samples. The genes that were differentially expressed in the 30 day IE group were compared with genes that were differentially expressed in the 60 day IE group. The venn diagram on the left depicts this information for Module M5 whereas the venn diagram depicts this information for Module M46.

The Figure 5.4 shows the genes that were upregulated and downregulated in IE samples compared to NE samples. The genes that were differentially expressed in the 30 day IE group were compared with genes that were differentially expressed in the 60 day IE group. The venn diagram on the left depicts this information for Module M5 whereas the venn diagram depicts this information for Module M46.

For example, TNFR1 protein can trigger either inflammation or apoptosis (self-destruction of the cell). Signaling within the cell initiates a pathway that turns on a protein called nuclear factor kappa B, which triggers inflammation and leads to the production of immune system proteins called cytokines. Apoptosis is initiated when the TNFR1 protein forms a complex with another protein and enters the cell to start a process known as the caspase cascade. Mutations of TNFR1 cause misfolded proteins that get trapped within a cell and constantly trigger alternative inflammation initiation pathways which lead to excess inflammation. The fourth gene NCF1; NCF1C plays a role in adjusting the inflammatory response. This gene encodes for proteins that form the NADPH oxidase complex which regulates the activity of phagocytes and neutrophils and those cells are known to play a role in adjusting the inflammatory response to optimize healing and reduce injury to the body. The remaining two genes differentially expressed in the 30 day IE samples are involved in the drug response mechanism and lung development. The gene SLC38A is an acetyltransferase which encodes proteins that activate and deactivate certain drugs and carcinogens. Polymorphisms in this gene are responsible for the polymorphism of N-acetylation which separates humans into rapid, inter-
mediate and slow acetylator phenotypes. This is interesting as it might indicate an association with the different exacerbation phenotypes as this gene is differentially expressed between IE and NE in the 30 day samples but not the 60 day samples. The gene KLF2, Krppel-like Factor 2, encodes a protein that has been associated with multiple biochemical processes in the human body, including lung development, epithelial integrity, T-cell viability, and adipogenesis.

Three of the genes involved in immune system functioning, IGF2R, PPP6C, and NCF1; NCF1C were also differentially expressed in the 60 day IE samples as was KLF2. Of the additional seven genes differentially expressed in the 60 day IE samples, two of them, LSP1 and TSC22D3, are also involved in the anti-inflammatory response, and the remaining five encode for proteins required for various functions such as structural components of the extracellular matrix, cell cycle regulation, and platelet function and release.

We also performed a gene set enrichment analysis using the MSigDB gene sets that are available for download from the broad institute [35]. These gene sets are divided into 7 collections of which we use one: curated gene sets. The curated gene sets are further organized by into sets that come from online pathway databases, publications in PubMed, and knowledge of domain experts. We selected the gene sets curated from the KEGG pathway database. KEGG [24, 25] gene sets include 186 gene sets derived from the KEGG pathway database which is a collection of manually drawn pathway maps and represent knowledge on the molecular interactions and reaction networks for metabolism, genetic information processing, environmental information processing, cellular processes, organismal systems, and human diseases. Our gene set enrichment analysis included a hypergeometric overlap of the genes in our module and the gene sets mentioned above. We use an adjusted p-value cut-off of 0.05 to select significant overlaps.

We also performed an enrichment using the Gene Ontology file which we downloaded from the Gene Ontology Consortium [3]. The GO project has developed three structured, controlled vocabularies (ontologies) that describe gene products in terms of their associated biological processes, cellular components and molecu-
lar functions. GO is structured as a graph, where each GO term is a node, and the edges represent relationships between the terms. GO is a hierarchy, where child terms are more specialized than their parent terms, and a term may have more than one parent term [3]. We filtered the ontology to use only the biological processes category. Biological Process terms represent operations or sets of molecular events with a beginning and end, which are relevant to the functioning of cells, tissues, organs, and organisms. From the biological processes, we selected only those categories containing more than 175 genes. We further filtered those to only retain the categories where each of their children contains less than 175 genes. These are considered to be the informative functional categories as described in [43]. We performed another hypergeometric overlap of the genes in our module and the genes in the informative functional categories. We used an adjusted p-value cut-off of 0.05 to select significant overlaps. The results of these enrichment analyses are presented in Figure 5.5.

5.2 Baseline module functional annotation

In an attempt to reveal the functionality associated with our baseline modules we performed enrichment analyses on all 53 modules using gene sets representing different tissues, pathways and gene ontologies. The results of those analyses are presented here.

5.2.1 Tissue enrichment

In order to determine whether our baseline modules may be specific to certain cell types, we performed a tissue specific enrichment using the Gene Set Enrichment Profiler by Benita et al [6]. They provide enrichment values derived by applying LiMma pairwise to hundreds of tissue expression profiles. LiMma generates coefficients as a measure of the difference between two groups and the sum of the coefficients of comparisons with a p-value < 0.05 are used to derive an enrichment value for each gene. Using those enrichment values we performed a hypergeometric overlap of the genes in each module and the tissue specific enrichment values provided by Benita et al. This analysis didn’t reveal that our modules are specific
Figure 5.5: Module M5 enrichment in KEGG pathways and GO biological process. A hypergeometric overlap of genes in module M5 and genes in KEGG pathways and GO biological process categories was performed and a p-value cut-off of 0.05 was applied.

to cell types as can be seen in Figure 5.6 where the median enrichment of the genes in each of our modules are plotted and the tissues are grouped by cell type. We can see from this figure that there isn’t any one cell type that is highly enriched in any of the modules.

5.2.2 Pathway enrichment

We then performed a gene set enrichment analysis using the gene sets curated from the KEGG pathway database and the Gene Ontology file from the Gene Ontology Consortium, which we described in Section 5.1.4. Our enrichment analysis included a hypergeometric overlap of the genes in each of our baseline module and the genes in the KEGG pathway gene sets as well as the genes in the Gene Ontology informative functional categories mentioned in Section 5.1.4. We used an
Figure 5.6: Median enrichment values of the genes in each module. The different tissues are plotted along the x-axis and are grouped by tissue type. The 53 baseline modules are plotted along the y-axis.
adjusted p-value cut-off of 0.01 to select significant overlaps. The results are depicted in Figure 5.7 (KEGG) and Figure 5.8 (Gene Ontology). We can see from these results that although there was some overlap in the biological processes and pathways associated with each module, the modules were overall quite heterogeneous, as they were associated with different arrays of biological processes and kegg pathways.
Figure 5.7: KEGG pathways associated with our modules. Only modules that had at least one significant (adjusted p-value < 0.01) pathway association are shown.
Figure 5.8: Gene Ontology biological processes associated with our modules. Only modules that had at least one significant (adjusted p-value < 0.01) biological process associated with them are shown.
Chapter 6

Conclusion

In this thesis we studied the problem of identifying modules of genes that are involved in a common functional role. We first presented several clustering based methods as well as several reverse engineering approaches that have been used in a similar capacity. We then introduced our methodology which constructs modules of genes by identifying genes that are co-expressed across many diseases. We evaluated the quality of our module construction procedure using Jaccard distance as an objective measure. Using this measure we illustrated that we obtain modules with high concordance regardless of the order in which diseases are presented to the algorithm which is an improvement on previous methods. We then used our module construction procedure to construct baseline modules pertinent to the lung. These modules were constructed using 5 peripheral blood data sets, each of which contained control samples used in different studies involving the lungs. We assessed the stability of our module construction procedure by applying it to three different data sets, a COPD dataset, a COPD normal dataset and a lung tissue dataset. We used resampling techniques along with Jaccard distance, as an objective measure, to evaluate the concordance of modules constructed by our method within a dataset and between different data sets. We found that our method produces modules with high concordance both within datasets and across different datasets as indicated by high Jaccard2 distances and small ranges of variation. We also assessed the sensitivity of our procedure to sample size and module size and found that our procedure produces more stable modules when sample size is larger.
and using a smaller sample size produces a larger number of modules with a larger variation in the number of modules produced. Furthermore we used our baseline modules in a case study using COPD samples and demonstrated the benefits of using modules in biomarker discovery. We performed an enrichment analysis using a list of genes that were previously shown to be differentially expressed between two phenotypes of COPD and identified modules that were enriched in that list. We then used the enriched modules and COPD samples that were stratified into three sub-phenotypes to compute the first principal component, which can be thought of as an eigen-gene, and projected the stratified samples onto that space. This experiment identified modules that could lead to biomarkers that may be used to stratify patients into different phenotypes upon diagnosis and help tailor the treatment to each individual. Finally we performed a differential expression analysis, by module, using moderated t-tests and compared the resulting list of differentially expressed genes to previously identified differentially expressed genes and found that by using our modules we can identify a larger number of differentially expressed genes as the false discovery rate is decreased.
Bibliography

[1] Cost risk analysis for chronic lung disease in canada -


70


