NEW CLUSTERING AND FEATURE SELECTION PROCEDURES
WITH APPLICATIONS TO GENE MICROARRAY DATA

by

YAOMIN XU

Submitted in partial fulfillment of the requirements
For the degree of Doctor of Philosophy

Dissertation Advisor: Dr. Jiayang Sun

Department of Statistics
CASE WESTERN RESERVE UNIVERSITY

January 2008
CASE WESTERN RESERVE UNIVERSITY

SCHOOL OF GRADUATE STUDIES

We hereby approve the dissertation of

YAOMIN XU

candidate for the Doctor of Philosophy degree

Committee Chair: 
Dr. Jiayang Sun 
Dissertation Advisor 
Professor, Department of Statistics

Committee: 
Dr. Joe Sedransk 
Professor, Department of Statistics

Committee: 
Dr. Sunil Rao 
Associate Professor, Department of Epidemiology & Biostatistics; 
Adjunct Associate Professor, Department of Statistics

Committee: 
Dr. Joe Nadeau 
Professor, Department of Genetics

Committee: 
Dr. Stephen Ganocy 
Assistant Professor, Department of Epidemiology & Biostatistics; 
Director, Data Management & Statistical Analysis Unit, Bipolar Disorders Research Center
# Table of Contents

Table of Contents ........................................ iii
List of Tables ............................................. vi
List of Figures ............................................... vii
Acknowledgement ........................................... x
Abstract ...................................................... xi

## 1 Introduction

1.1 Microarray Technique .................................. 2
  1.1.1 Microarray Platforms ................................. 4
  1.1.2 Microarray Applications ............................. 7
  1.1.3 Image Analysis ...................................... 10
  1.1.4 Preprocessing ....................................... 11
  1.1.5 Microarray Data Analysis ......................... 12

1.2 Cluster Analysis ....................................... 15
  1.2.1 Measure of Proximity ................................ 16
  1.2.2 Hierarchical Clustering ............................. 18
  1.2.3 Partitioning ........................................ 21
  1.2.4 Other Clustering Algorithms ....................... 23

1.3 Feature Selection ..................................... 25
  1.3.1 Introduction ....................................... 25
  1.3.2 Feature Selection in Microarray Data Analysis .. 26

1.4 Road Map of the Following Chapters ................. 28

## 2 PfCluster: A New Profile Cluster Analysis Procedure 30

2.1 Introduction ........................................... 30
  2.2 Profile Patterns ..................................... 32
    2.2.1 Parallel Case .................................. 33
    2.2.2 Other Cases .................................... 34

2.3 Dissimilarity Matrix .................................. 35

2.4 Coherence Index ...................................... 36

2.5 Threshold for Coherence Index ....................... 37
2.5.1 The First Order Approximation: \( d_1^* \) ........................................... 38
2.5.2 The Second Order Approximation: \( d_2^* \) ........................................... 39
2.6 Estimation of \( \Sigma_s \) ................................................................. 49
  2.6.1 Estimation of \( \Sigma_x \) ......................................................... 49
  2.6.2 Dendrogram Cut-value \( h \) for Estimating \( \Sigma_x \) ..................... 50
2.7 Distribution of \( D_{ii}^{2} \) for Positive Semidefinite \( \Sigma_s \) .......... 51
2.8 PfCluster .................................................................................. 52
  2.8.1 Preprocessing ..................................................................... 52
  2.8.2 Clustering Algorithm ............................................................ 56
2.9 Simulation Studies ...................................................................... 59
  2.9.1 Study I ............................................................................... 59
  2.9.2 Study II ............................................................................. 66
2.10 Real Data Applications ................................................................. 71
  2.10.1 Biomarker LOH data ............................................................ 71
  2.10.2 Folate Acid Data ................................................................. 78
2.11 Comparison with Correlation Coefficient Based Approach ....... 85
  2.11.1 Cophenetic Correlation ........................................................ 87
2.12 Discussion .................................................................................. 90

3 RPselection: A New Feature Selection Procedure .................. 92
  3.1 Introduction ............................................................................. 92
  3.2 Methods .................................................................................. 95
    3.2.1 Partition Clustering Procedure \( g(\cdot) \) ................................ 96
    3.2.2 Correspondence Measure \( \phi \) .......................................... 98
    3.2.3 Criterion Function \( J(x) \) .................................................. 102
    3.2.4 Feature Selection .............................................................. 104
    3.2.5 RPselection Algorithm ...................................................... 106
    3.2.6 Choice of \( d, B \) and \( S \) .................................................. 107
  3.3 Simulation Studies .................................................................... 113
    3.3.1 Hierarchical Error Model (HEM) ..................................... 114
    3.3.2 Simulated Data ................................................................. 115
    3.3.3 Performance Measures ..................................................... 117
    3.3.4 Simulation Results .............................................................. 119
  3.4 Real Data Application ................................................................. 124
    3.4.1 Uveal Melanoma Data ........................................................ 124
    3.4.2 Results ............................................................................. 127
  3.5 Discussion .................................................................................. 131
List of Tables

2.1 Compare cophenetic correlation coefficient for PfCluster and correlation coefficient based clustering using the simulated data I ................................................................. 88
2.2 Compare cophenetic correlation coefficient for PfCluster and correlation coefficient based clustering using the simulated data II .............................................................. 89
2.3 Compare cophenetic correlation coefficient for PfCluster and correlation coefficient based clustering using LOH data ................................................................. 89
3.1 Power $q_{F_{IT}}$ of the RPselection algorithm ......................... 121
3.2 IQR of $q_{F_{IT}}$ from 25 repetitions ........................................ 122
3.3 Average and standard deviation of $q_{F_{IT}}$, for $n = 15$ .......... 123
3.4 Average and standard deviation of $q_{F_{IT}}$, for $n = 30$ .......... 123
3.5 Average and standard deviation of $q_{F_{IT}}$, for $n = 45$ .......... 123
3.6 $q$ values from the uveal melanoma data .......................... 135
3.7 $R$ values from the uveal melanoma data .......................... 136
A.1 Lance-Williams parameters for standard hierarchical clustering methods .................................................. 178
List of Figures

1.1 A typical cDNA microarray image .......................... 5
1.2 cDNA Microarray technology .............................. 6
1.3 PM and MM design in Affymetrix GeneChip .............. 6
1.4 Affymetrix GeneChip ........................................ 7
1.5 Illumina’s 3-micron beads in wells ........................ 8
1.6 Illumina Sentrix 96 fiber optic array matrix and beadchip 8
1.7 Dendrogram .................................................. 20

2.1 Motivating profile pattern ................................. 31
2.2 Meaningful profile pattern ................................. 32
2.3 Parallel profile with $t = 5$ ................................ 33
2.4 Comparison of $d^*, d_1^*, and d_2^*, s^2 = 0.01$ ........... 47
2.5 Comparison of $d^*, d_1^*, and d_2^*, s^2 = 0.05$ ........... 48
2.6 Dendrogram .................................................. 57
2.7 PfCluster dendrogram demo ............................... 58
2.8 Profile patterns of the simulated data I .................. 60
2.9 Data profiles of the simulated data I ...................... 61
2.10 Heatmap of the simulated data I ......................... 62
2.11 PfCluster dendrogram of the simulated data I .......... 62
2.12 Profile groups of the simulated data I recovered by PfCluster 63
2.13 Heatmap of the simulated data I with genes ordered by the PfCluster dendrogram ............................... 64
2.14 Profile patterns of the simulated data II ................ 66
2.15 Data profiles of the simulated data II ................... 67
2.16 Heatmap of the simulated data II ....................... 68
2.17 PfCluster dendrogram of the simulated data II ........ 68
2.18 Profile groups of the simulated data II recovered by PfCluster 69
2.19 Heatmap of the simulated data II with genes ordered by the PfCluster dendrogram ............................... 70
2.20 Data profiles of LOH data ................................. 72
2.21 Heatmap of LOH data ...................................... 72
2.22 PfCluster Dendrogram of LOH data ........................................ 73
2.23 Profile groups 1-24 of LOH data by PfCluster ...................... 74
2.24 Profile groups 25-38 of LOH data by PfCluster ..................... 75
2.25 Profile group 5 of LOH data by PfCluster ............................ 76
2.26 Heatmap of LOH data with markers ordered by the PfCluster 
dendrogram ................................................................. 77
2.27 Experimental Design of the folate acid data ......................... 79
2.28 Microarray reference design of the folate acid data ............... 79
2.29 Gene expression profiles of top 0.1 % of genes in Folate data ... 80
2.30 PfCluster dendrogram using the top 0.1% of genes in Folate 
data ............................................................................... 81
2.31 Resulting clusters, Group 1 - 6, from PfCluster using top 0.1% 
of genes in Folate data. ....................................................... 82
2.32 Resulting clusters, Group 7 - 12, from PfCluster using top 
0.1% of genes in Folate data. .............................................. 83
2.33 Resulting clusters, Group 13 - 1, from PfCluster using top 
0.1% of Folate data. ......................................................... 84
2.34 Compare PfCluster with correlation coefficient based clustering 
using the simulated data I ................................................. 86
2.35 Compare PfCluster with correlation coefficient based clustering 
using simulated data II .................................................... 86
2.36 Compare PfCluster with correlation coefficient based clustering 
using the LOH data .......................................................... 86

3.1 Illustration of patterns for 1-dimensional \( J_n(x) \) .................... 104
3.2 Illustration of patterns for 2-dimensional \( J_n(x) \) .................... 105
3.3 CV of \( n_x(d) \) vs d .......................................................... 108
3.4 Probability that each gene is selected at least once in the RPselect 
algorithm ................................................................. 111
3.5 \( \phi \) vs. \( d \) of RPselect algorithm ....................................... 112
3.6 The structure of the simulated data ....................................... 116
3.7 M vs A plot of the simulated data ....................................... 117
3.8 Dendrogram of the simulated Data ....................................... 118
3.9 Power \( q_{FIT} \), of the RPselect algorithm ............................. 121
3.10 IQR of \( q_{FIT} \) from 25 repetitions ..................................... 122
3.11 Boxplot of all the samples in the uveal melanoma data .......... 124
3.12 Correlation-based clustering using all 3514 genes on all the 
samples in the uveal melanoma data .................................. 125
3.13 Correlation-based multidimensional scaling using all 3514 genes 
on all the samples in the uveal melanoma data ................. 126
3.14 Performance score by applying the RPselection algorithm to the uveal melanoma data .................................................. 128
3.15 Fitness score \( FIT^2 \) for the uveal melanoma data ............... 129
3.16 Selected key genes for the uveal melanoma data .................... 130
3.17 Demonstration of \( J_n(x) \) of the uveal melanoma data over different sets of \( B \) and \( d \) .................................................. 131
3.18 Demonstration of \( FIT(x) \) of the uveal melanoma data over different sets of \( B \) and \( d \) .................................................. 132
3.19 Demonstration of the impact of \( B \) and \( d \) over \( J_n(x) \) on the uveal melanoma data .................................................. 133
3.20 \( q \) values from the uveal melanoma data .......................... 135
3.21 \( R \) values from the uveal melanoma data .......................... 136

4.1 Flow chart of software package pfclust .................................. 139
4.2 Heatmap of the biomarker LOH data after the MIP filtering .... 148
4.3 First step application of PfCluster to the filtered biomarker LOH data ................................................................. 149
4.4 M-vs-h plot from the 1st step application of PfCluster to the filtered biomarker LOH .................................................. 150
4.5 The improved 2nd step application of PfCluster to the filtered biomarker LOH .................................................. 151
4.6 The coherent cluster 1 in the improved PfCluster dendrogram (top) and its profiles (bottom) ................................. 153
4.7 Profiles of the coherent clusters 1-4 in the improved PfCluster dendrogram .................................................. 154
4.8 Heatmap of the input data with biomarkers ordered by the improved PfCluster dendrogram .................................. 155
4.9 Flow chart of software package rpselect ................................ 156
4.10 Demonstration of plot method for \( J_n(x) \) .............................. 162
4.11 Demonstration of plot method for \( FIT(x) \) .............................. 163

5.1 Feature selection with RPselection ...................................... 169
5.2 Profile cluster analysis and visualization with PfCluster ........ 170
5.3 Coherent groups from PfCluster ........................................ 171
5.4 Approximate \( d^* \) using \( d^*_2 \) ........................................ 173
ACKNOWLEDGEMENTS

First of all, I would like to express my sincere gratitude to my advisor, Dr. Jiayang Sun, for her immense help at every stage of my research. I remain grateful for her constant encouragement, and mental support throughout my graduate study at Case Western Reserve University. Without her patience, guidance and encouragement, none of this work would have been possible. As a mentor, her wisdom, kindness and enthusiasm benefitted me greatly in both my research work and life. Her valuable insights and ideas directly and significantly contributed to the work in this dissertation.

I also extend my gratitude to Dr. Joe Sedransk, Dr. Joe Nadeau, Dr. Sunil Rao, and Dr. Stephen Ganocy for serving on my committee. I appreciate their suggestions, valuable discussion, and precious time. I thank all the other faculty and students in the statistics department for their help throughout my graduate study.

A special thanks goes to Dr. Stephen Ganocy, Dr. Li Zhang and Mr. Jeff Hammel for proof-reading the manuscript and providing helpful suggestions. All remaining errors are mine.

I would like to convey my appreciation to Dr. Charis Eng, Dr. Joe Nadeau, Dr. Toshimori Kitami, and Dr. Arun Singh for providing data and giving valuable suggestions during my research.

I would also like to thank my colleagues and co-workers at the Cleveland Clinic Foundation for their long-term support to both my work and my study.

Last, but not the least, my sincere thanks goes to my wife Meng and my daughter Rina for their endless love, continuous support and encouragement. My appreciation goes to my parents Shizhen Xu, and Litai Gu, and my parents-in-law Boqin Xu, Yanping Wang for their dedication and many years of support for now and future. This work is dedicated to them.
New Clustering and Feature Selection Procedures
with Applications to Gene Microarray Data

Abstract

by

Yaomin Xu

Statistical data mining is one of the most active research areas. In this thesis
we develop two new data mining procedures and explore their applications
to genetic data.

The first procedure is called PfCluster - Profile Cluster Analysis. It is
a clustering method designed for profiled genetic data. The PfCluster is
efficient and flexible in uncovering clusters determined by a new class of bi-
ologically meaningful distance metrics. A new internal quality measure of
clusters, coherence index, is developed to find coherent clusters. An efficient
mechanism for choosing the threshold of coherent clusters is also derived and
implemented. The threshold is based on the first and second order approxi-
mations to the true threshold under a null distribution for parallel clusters.
The PfCluster has been applied to simulated data and two real data exam-
pies: a biomarker LOH dataset and a microarray gene expression dataset.
PfCluster is competitive to the correlation-based clustering procedures.

The second procedure is called RPselection - Resampling based parti-
tioning selection. It is a feature selection algorithm designed for microarray
studies. It selects a subset of genes that maximizes a fitness score. The fitness
score measures the relevance between the partition labels from a clustering
result and an external class label derived from the clinical outcomes. The score is computed using a resampling procedure. The RPselection algorithm has been applied to simulated data and a real uveal melanoma gene expression data. RPselection outperforms gene-by-gene test-based feature selection procedures.

Software development is an integral part of modern statistical research. Two software packages, *pfclust* and *rpselect*, are developed in this thesis based on our PfCluster method and RPselection algorithm. Packages *pfclust* and *rpselect* are implemented based on R object-oriented programming framework, and they can be easily customized and extended by users.

The ideas in our two procedures can be generalized and applied to other data mining tasks. This thesis concludes with discussion on connections between two methods and the related future research.

**Key words:** Bioinformatics, coherence index, data mining, feature selection, gene expression pathway, gene profiling, informative gene, microarray data, profile cluster analysis, partitioning, regulatory network, statistical pattern recognition
Recent development of microarray technologies has highlighted the need for new data mining techniques to mine and discover ‘biologically meaningful’ information from massive amounts of genetic data. A typical microarray experiment may have a small number of records (less than 100), while the number of features (i.e. variables or genes) are in tens of thousands. This \textit{large p small n} setting of microarray data analysis creates a high likelihood of finding “false positives”, so we need robust methods to analyze microarray data. The main types of data analysis performed for microarray data include \textit{gene selection}, \textit{classification}, and \textit{clustering}.

Gene selection is a process that finds the genes most strongly related to a particular sample class. The sample class can be a cell class; a tumor class; a disease class; a cancer class; a tumor stage, such as an onset, early, middle, or late stage; or simply any biologically meaningful class. Identifying such genes is helpful to find a cure for a disease or to understand the development of a cancer.

Gene classification is a supervised learning technique that is built on both gene expression data and their sample classes. It classifies the disease class or predicts the outcome of a sample based on its gene expression pattern.
For example, based on gene expression profiles, breast tissue specimens may be classified as either normal or cancerous; patients may be classified into responders and non-responders to a treatment.

Gene clustering is an unsupervised learning technique that is built on gene expression data only. It is especially useful when subtypes of certain diseases are clinically indistinguishable but are genetically heterogeneous. Hence, based on gene expression profiles a correct clinical diagnosis may be made quickly, which is crucial for selecting a proper medical therapy. Gene clustering can be also used to discover new biological classes or refine existing sample classes.

This thesis provides two new data mining procedures, PfCluster for clustering and RPselection for gene selection of microarray data. In the rest of this chapter, we provide the background and preliminaries for this thesis. In § 1.1, we briefly introduce microarray technology, and review preprocessing procedures and basic data analysis techniques of microarray data. In § 1.2, we review basic cluster analyses and feature selection techniques, as well as the background for our two new data mining tools.

1.1 Microarray Technique

A microarray is a biological assay, with an ordered array of nucleic acids, proteins, or small molecules, that enables the parallel analyses of complex biochemical samples (Schena et al., 1995). DNA microarrays provide a method for exploring the genome\(^1\) of an organism at molecular level. Our present knowledge, in most (if not all) genetics textbooks, of how genes are regulated stems from the analysis of a limited number of genes. Microarrays herald a new area of investigation of gene regulation that promises to provide much deeper understanding of how cells coordinate the expressions of thousands of

\(^1\)The genome is all of the genetic information or hereditary material contained in the DNA (or, for some viruses, RNA) of an organism.
genes. We now can identify all the genes of an organism that change their expressions under different conditions, and hope to understand the mechanism of a cell’s response to a particular condition. Such information provides key clues to the function of individual proteins.

A microarray consists of a solid surface on which strands of polynucleotides\(^2\) have been attached in specified positions. We refer to the polynucleotides immobilized on the solid surface as probes. Probes consist of either of cDNA\(^3\) printed on the surface or shorter oligonucleotides\(^4\) synthesized/deposited on the surface. Labels\(^5\) may be tagged to target the bindings by hybridization to the probes on the array with which they share sufficient sequence complementarity. After sufficient time for the hybridization reaction, the excess sample is washed off from the solid surface. At that point, each probe on the microarray should be bound to labeled targets whose quantity is proportional to the expression level of the gene represented by that probe. By measuring the intensity of the label bound to each probe, one obtains the numbers which, after adjusting for technical artifacts, should provide an estimate of the expression levels of all the corresponding genes. For further details see Duggan et al. (1999).

DNA Microarray technology makes use of the sequence information available from the genome projects and other sequencing efforts to answer questions regarding the genes that are expressed, for example, in a particular cell type of an organism, at a particular time, and under a particular condition. We call those gene expression measurements gene profiles or patterns. Using those profiles/patterns, biologists often seek to know, for example, which

\(^2\)A polynucleotide molecule is an organic polymer molecule comprised of nucleotide monomers covalently bonded in a chain. DNA (deoxyribonucleic acid) and RNA (ribonucleic acid) are examples of polynucleotides with distinct biological function.

\(^3\)In genetics, complementary DNA (cDNA) is DNA synthesized from a mature mRNA template in a reaction catalysed by the enzyme reverse transcriptase.

\(^4\)Oligonucleotides are short sequences of nucleotides (RNA or DNA).

\(^5\)A label refers to a specific DNA sequence tagged or labelled so that it can be independently identified.
genes are co-regulated under a set of conditions, which genes are co-regulated by a common transcription factor, which genes of unknown function are co-regulated as genes with known functions, or in which regulatory networks the gene is involved.

Microarray technology, as a high throughput approach of differential gene expression studies, efficiently generates massive amount of gene regulation data, facilitating scientists to quickly identify which gene candidates to follow up with functional characterization.

1.1.1 Microarray Platforms

Microarrays differ in many important details. Here we give a brief introduction on three popular platforms.

Two Color cDNA Microarray

The cDNA microarrays usually consist of probes of cDNA robotically printed on a microscope slide coated with poly-lysine or poly-amine to enhance absorption of DNA probes (Schena et al., 1995). The robotic printers have several pins arranged in a rectangular pattern. For example, for a four-pin printer, the spots on the array are printed in four rectangular grids corresponding to the rectangular arrangement of the robotic pins (Figure 1.1). Because the cDNA probes are generally several hundred bases long, stringent hybridization conditions are often employed and nonspecific hybridization is limited. Since robotic printing varies substantially in size and shape of corresponding spots on different arrays, direct comparison of intensities of corresponding probes on different arrays is problematic. Much of this array-to-array variability can be controlled by co-hybridizing two samples on the same array. In current technology, two cDNA samples on the same array are labeled with different fluorescent dyes. By using two laser sources, the intensity of fluorescence in each of the two frequency channels is measured.
Figure 1.1: A typical cDNA microarray image (Hegde et al., 2000)

simultaneously at each probe. The differences or ratios of the differential expression are then computed (see Figure 1.2).

Affymetrix GeneChip Arrays

The technology for the production of high-density oligonucleotide arrays (Lockhart et al., 1996) was pioneered by Affymetrix. In this type of array, a gene is represented by a set of 25 or so oligonucleotides, called perfect match probes (PM). Each perfect match probe is paired with an artificially created mismatch probe (MM) that is formatted by changing the middle base of the corresponding perfect match probe to its complementary base (Figure 1.3). The mismatch probe is intended to act as an internal control. The hybridization of the gene to the perfect match probe represents specific hybridization. The bindings of the specific hybridization should be stronger than those of any nonspecific hybridization to the mismatch probe. Affymetrix refers to each PM-MM pair as a probe pair and the entire set of probe pairs for a gene is called a probe set (Figure 1.3). High-density oligonucleotide microarrays are manufactured by synthesizing the oligonucleotides directly onto the
surface of a silicon chip. Data from an experiment show the expression of thousands of genes on a single GeneChip (Figure 1.4)

**Bead-based Arrays**

New technologies are constantly emerging in an effort to extend the throughput and the potential of microarrays. One of the new platforms is bead-based fiber-optic microarray technology (Walt, 2000). The fiber-optic array
uses randomly ordered, self-assembled arrays of beads that provide a platform for parallel analysis of complex biological samples (Walt, 2000; Michael et al., 1998). Arrays of beads are randomly assembled onto patterned optical fiber bundles. Individual bead has a diameter of 3 microns, and beads are assembled into a single array (Figure 1.5). A decoding process is used to specify the identity of each bead in each array location. Each bead contains oligonucleotide probes which hybridize with high specificity to complementary sequences in a complex nucleic acid mixture. The miniaturized fiber optic arrays are further built into a 96- or 384-array matrix that matches microtiter plates. This allows the processing of many (96 or 384) samples simultaneously in an automated fashion (Figure 1.6). Since each array can interrogate hundreds to over one thousand targets, a 96- or 384-array matrix will allow hundreds of thousands of assays to be carried out rapidly and efficiently (Oliphant et al., 2002). Such a format is suitable for assaying patient samples in a hospital setting.

1.1.2 Microarray Applications

Microarray techniques have already been heavily used in biological research to address a wide variety of questions. Listed below are some examples for
Figure 1.5: Illumina’s 3-micron beads in wells (Wadman, 2006)

Figure 1.6: Illumina Sentrix 96 fiber optic array matrix and beadchip (Gershon, 2004)
common uses of microarray experiments.

**Functional Genome Analysis**

DNA Microarray analysis allows comparisons between the expression levels of certain genes across different tissues and pathological conditions. Microarray technology reveals the temporal and spatial patterns of expressions of all the genes involved in the developing processes of an organism. Exploring patterns of gene expressions by microarray analysis can provide us with functional clues for dissecting complicated processes in a whole organism and in disease states.

**Disease Diagnosis**

Microarray analysis will improve our understanding of diagnosis and prognosis. Transcription\(^6\) profiling using DNA microarrays has great potential as a systematic approach for discovering new classes of tumors and for assigning known tumors to classes to predict response to therapy. Gene expression monitoring could provide new insights into many aspects of tumor pathology, including cell of origin, stage and grade of a tumor, clinical course of a treatment, and patient’s response to a treatment. Other applications include identification of targets for drug development, disease diagnosis and prognosis, gene copy number\(^7\) detection, risk assessment, gene mutation, and Single Nucleotide Polymorphism (SNP) analysis.

---

\(^6\) Transcription is the process through which a DNA sequence is enzymatically copied by an RNA polymerase to produce a complementary RNA. It is the transfer of genetic information from DNA into RNA. In the case of protein-encoding DNA, transcription is the beginning of the process that ultimately leads to the translation of the genetic code (via the mRNA intermediate) into a functional protein.

\(^7\) Gene copy number (also “copy number variants” or CNVs) is the number of copies of a particular gene in the genotype of an individual.
Pharmacogenomics

Pharmacogenomics is the study of how an individual’s genetic inheritance affects his/her body’s response to drugs. Microarray experiments have been used to exploit functional genomics and molecular toxicology for obtaining correlations between therapeutic responses to drugs and the genetic profiles of patients. Microarrays are accelerating drug discovery and testing.

Toxicogenomics

Toxicogenomics combines studies of genetics, mRNA expression, cell and tissue-wide protein expression and metabolomics to understand the role of gene-environment interactions in disease.

Environmental Monitoring

Environmental factors are known to affect gene expressions. It is important to assess the genome-level impact of exposure to environmental stressors, especially contamination of air, food, or water. Microarrays can be used to compare the gene expression patterns across different environmental factors.

1.1.3 Image Analysis

When microarrays are scanned at the end of an experiment, the result is a series of images, one image per channel. Thus a single-channel microarray, such as an oligonucleotide array, yields one image per array, whereas a two-channel microarray yields two images per array, one image per channel. In this process, the scanner reads in a microarray by dividing it into a very large number of pixels and recording the intensity level of the fluorescence at each pixel. The resulting rectangular array of pixels and their associated intensities constitutes the image of the microarray. The image must be converted into spot intensities for analysis. The purpose of this conversion is to
assign every DNA sequence spotted on the microarray an intensity measure which reflects the amount of labeled samples that hybridized to it. The raw data produced from microarray experiments are the hybridized microarray images. To obtain information about gene expression levels, these images should be analyzed first.

A variety of software tools have been developed for use in image analysis. The basic idea is to obtain the varying intensities for each spot in an array. Although this is a relatively straightforward goal, there is currently no common manner of extracting this information. Many research groups are still writing customized software for this purpose. Scanning and processing images are currently resource-intensive tasks, requiring human intervention to ensure that grids are properly aligned and that artifacts are flagged and properly excluded from subsequent analyses.

Image analysis involves three stages. First, arrayed genes must be identified from spurious signals that arise due to precipitated probe, other hybridization artifacts, or dust on the surface of the slide. After gridding, the spot intensities (real signal) and background (noise) have to be calculated for spots. Background is calculated either locally for each spot, or globally for the entire image. The third step in image processing is the extraction of signal from noise and quality control for the spot. More details can be found in Bassett et al. (1999) and Hegde et al. (2000)

1.1.4 Preprocessing

Once an experiment has been run and spot intensity data has been collected, it is necessary to preprocess the data prior to its analysis. The following tasks often need to be accomplished in the data preprocessing stage:

- Transform the data into a scale suitable for analysis,
- Normalize the data to remove the effects of systematic sources of vari-
Data transformation and normalization are typically required for preprocessing. Logarithmic transformation or the generalized logarithmic transformation (Durbin and Rocke, 2004; Huber et al., 2002) is usually applied to expression data after image processing. The log-transformation is preferable because: 1) the variation of log-transformed data is less dependent on the magnitude of the values for most data; 2) it reduces the skewness of highly skewed data; 3) it stabilizes the variance and makes variance estimation easier; and 4) it facilitates the visualization of the data. Often base-2 logarithm is used. Normalization is applied to adjust the individual hybridization intensities obtained under different technical conditions in order to balance them appropriately for meaningful biological comparisons. The different technical conditions include unequal starting RNA quantities, different labeling or detection efficiencies between the fluorescent dyes, and systematic biases in measured expression levels. The commonly used normalization procedures are Total Intensity Normalization, Mean/Median Normalization, Quantile Normalization, and LOWESS Normalization (Quackenbush, 2002). Filtering or quality checking includes signal to noise ratio checking, detection limits screening or low intensity filtering, and flip dye replicates checking.

1.1.5 Microarray Data Analysis

Two-Group Comparative Experiments

Comparative experiments are commonly used and have been performed in many microarray studies. The objective of this type of experiment is to compare the expression levels of genes across two or more conditions, particularly to identify genes that are significantly differentially expressed across these conditions. Common statistical approaches used for two-group comparisons include...
are $t$-test, other variants of $t$-test, a permutation test based on $t$ statistics, and the Mann-Whitney-Wilcoxon rank sum test. Since here we perform a very large number of statistical tests, it is necessary to overcome the multiplicity issues in performing many tests simultaneously. The common multiplicity adjustments are single-step *family-wise error rate* (FWER) adjustments, such as *Bonferroni* adjustment and *Sidak* (Sidak, 1967) adjustment; step-wise FWER controls, such as Holm-Bonferroni, Holm-Sidak (Holm, 1979), Hochberg (Hochberg, 1988), and Westfall-Young adjustments (Westfall and Young, 1993a); and the popular false discovery rate (FDR) adjustment (Benjamini and Hochberg, 1995; Yukieli and Benjamini, 1999). The two-group comparison is the simplest experiment. Its corresponding analysis techniques have been used widely in microarray studies. It is very effective and computationally efficient, but has the drawback of treating genes in isolation and leading to a high false positive rate (Section 3.1).

**Model Based Inference for Designed Experiments**

Linear model based approaches are applied to more complex microarray data analyses. Kerr and Churchill (2001c), Kerr and Churchill (2001b), and Kerr and Churchill (2001a) provided early work that applied linear models to the analysis of data from multichannel cDNA microarrays. Churchill (2002) and Yang and Speed (2002) gave a nice review of using linear model in microarray data analysis. Wolfinger et al. (2001) proposed a two-stage approach for fitting linear models, including mixed effect models. Chu et al. (2002) discussed linear models for oligonucleotide array experiments. This type of modeling provides a general constructive framework for complex experimental design on microarrays. However, the linear model approach still does not take into consideration of gene-by-gene interactions.
Pattern Discovery

The procedures described in the last two paragraphs may be called test-based approaches, which are conducted on a gene-by-gene basis. For experiments with a possible multi-gene interaction, it is useful to apply multivariate analysis techniques. Cluster analysis is one of those multivariate techniques for pattern discovery of microarray data. Among the cluster analysis approaches, hierarchical clustering is a class of the most widely used clustering methods. Partitioning methods form another popular class of clustering procedures. They split data into a specified number of non-overlapping clusters. We will provide more details about clustering techniques in § 1.2. An useful pattern discovery approach that extends the regular cluster analysis techniques is the two-way clustering, in which genes (row variables) and samples (column variables) are clustered respectively and cross linked internally. Other multivariate analysis techniques used in the literature of microarray data pattern discovery include Principle Component Analysis (Raychaudhuri et al., 2000; Yeung and Ruzzo, 2001), Factor Analysis, Multidimensional Scaling, and Projection Pursuit.

Classification

Statistical classification techniques are supervised learning techniques. They use samples and their known class information to build a rule that can be used to classify or predict the class of a new sample. In the context of microarray experiments, we would like to classify mRNA samples or other samples based on their gene expression profiles. The application of classification to gene microarrays can be found in Golub et al. (1999) and Hedenfalk et al. (2001). The statistical techniques for classification include linear discriminant analysis (LDA) (Fisher, 1936), recursive partitioning (Morgan and Sonquist, 1963), CHAID (Hartigan, 1975a), FIRM (Hawkins and for Math-
Clustering methods can be classified into two main categories: 1) hierarchical, and 2) non-hierarchical or partitioning methods, both encompassing a variety of algorithms.

Hierarchical methods produce a tree-like structure called a dendrogram, which illustrates hierarchies among the objects. These hierarchical methods are typically further subdivided into two categories: (a) agglomerative

---

8In some computer science literature some authors many also call (incorrectly) a clustering technique a classifier.
methods and (b) divisive methods.

Agglomerative hierarchical methods begin with each of the \( n \) objects in its own separate cluster and the dendrogram is generated by iteratively merging clusters until all objects are in the same cluster. The most popular agglomerative hierarchical methods are single linkage clustering, complete linkage clustering, average linkage clustering, and Ward’s minimum variance method. Divisive hierarchical methods start with all objects in a single cluster and the dendrogram is produced by splitting clusters until each object is in its own individual cluster.

Non-hierarchical or partitioning methods produce a separation of objects into \( K \) distinct clusters. The most commonly used partitioning procedure is the popular \( k \)-means algorithm, which efficiently produces a good partition solution to large-scale problems, but does not guarantee that a globally optimal solution will be obtained. \( K \)-medoids (Kaufman and Rousseeuw, 2005) is a robust version of \( k \)-means.

### 1.2.1 Measure of Proximity

A cluster analysis groups objects together based on how close they are to each other. The closeness is measured by the proximity, or the similarity, or reversely the distance between two objects. Here we describe several commonly used distance measures in the gene expressing setting. Readers can refer to Gordon (1999a) for a discussion of other distance measures.

Let \( x \) and \( y \) denote two objects in an experiment which are measured on \( p \) features. Then we define

- Euclidean distance:

\[
d(x, y) = \sqrt{\sum_{i=1}^{p} (x_i - y_i)^2}
\]  

(1.1)
Manhattan distance:
\[
d(x, y) = \sqrt{\sum_{i=1}^{p} |x_i - y_i|}
\]  
(1.2)

Pearson’s correlation coefficient:
\[
s(x, y) = \frac{\sum_{i=1}^{p}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{p}(x_i - \bar{x})^2 \sum_{i=1}^{p}(y_i - \bar{y})^2}}
\]  
(1.3)

where \( \bar{x} = \frac{1}{p} \sum_{i=1}^{p} x_i \), and \( \bar{y} = \frac{1}{p} \sum_{i=1}^{p} y_i \).

Spearman correlation coefficient:
\[
s(x, y) = \frac{\sum_{i=1}^{p}(r(x_i) - \bar{r}_x)(r(y_i) - \bar{r}_y)}{\sqrt{\sum_{i=1}^{p}(r(x_i) - \bar{r}_x)^2 \sum_{i=1}^{p}(r(y_i) - \bar{r}_y)^2}}
\]  
(1.4)

where \( \bar{r}_x = \frac{1}{p} \sum_{i=1}^{p} r(x_i) \), \( \bar{r}_y = \frac{1}{p} \sum_{i=1}^{p} r(y_i) \), \( r(x_i) \) and \( r(y_i) \) are the ranks of \( x_i \) and \( y_i \), respectively.

Euclidean distance is one of the most commonly used distance measure. However, for gene expression data the overall shapes of gene expression patterns (or profiles) are of greater interest than the individual magnitudes of each feature. Euclidean distance does not scale well for shifted and scaled patterns (profiles). Therefore, data standardization is sometimes applied before calculating the Euclidean distance between two objects.

Pearson’s correlation measures the similarity between the shapes of two expression profiles. Pearson’s correlation coefficient treats each object as a random variable with \( p \) observations and measures the similarity between two objects by calculating the linear relationship between the distribution of the two corresponding random variables. Pearson’s correlation is widely used and has been proved effective as a similarity measure for gene expression data (Jiang et al., 2003; Tang et al., 2001; Tang and Zhang, 2002; Yang and
Speed, 2002). However, empirical study has shown that it is not robust to outliers (Heyer et al., 1999) nor to non-scaled feature vectors. Spearman’s correlation is similar to Pearson’s correlation except that the ranks of the data are used instead of the observed values of $x$ and $y$. Spearman’s correlation is robust to outliers, non-normal distributions, and deviation from linearity in the relationship between $x$ and $y$. However, it only uses the rank information of the data.

### 1.2.2 Hierarchical Clustering

Hierarchical clustering proceeds successively either by merging smaller groups into larger ones or by splitting larger groups into smaller ones. The methods are then either agglomerative or divisive, respectively.

**Agglomerative hierarchical clustering**

Agglomerative clustering is processed in a bottom-up fashion. It is initiated with each gene as a cluster on its own. In the subsequent steps the closest pair of clusters are agglomerated into one cluster. The process can be continued until all the data fall into one cluster. The dendrogram is generated according to the iteratively merging of the subclusters.

In the process of the agglomerative clustering, the distance between two clusters needs to be re-defined. Listed below are commonly-used between cluster distances:

- **Single Linkage:** the distance between 2 clusters $C_1$ and $C_2$, denoted as $D(C_1, C_2)$, is the distance between the two closest objects between two clusters:

  $$D(C_1, C_2) = \min_{i \in C_1, j \in C_2} (d_{ij})$$

  \hspace{1cm} (1.5)

  where $d_{ij}$ is the distance between objects $i \in C_1$ and $j \in C_2$. 
• Complete Linkage: the distance between 2 clusters is the largest distance of any of the two objects between two clusters:

\[ D(C_1, C_2) = \max_{i \in C_1, j \in C_2} (d_{ij}). \]  

(1.6)

• Average Linkage: the distance between 2 clusters is the average of the distances of all pairs of objects between the two clusters:

\[ D(C_1, C_2) = \text{avg}_{i \in C_1, j \in C_2} (d_{ij}). \]  

(1.7)

• Centroid linkage: the distance between 2 clusters is the distance between the cluster centers.

The complete linkage method minimizes the maximum within cluster distance, and hence it tends to find compact clusters but may overemphasize small differences between clusters. The single linkage method maximizes the minimum distance between clusters and hence it may exhibit a chain effect, which places two very dissimilar objects into one cluster merely because they are linked via a few intermediate observations. The average and centroid linkages are compromises between single and complete linkage methods, but they are not monotone transformations of the distances. However, the average and centroid methods do usually provide a number of small tight clusters that can be useful.

Divisive hierarchical clustering

Divisive clustering is initiated with all the genes placed together in one cluster. In the subsequent steps the loosest cluster is split into two. The process can continue until each gene is a cluster on its own. Bi-partitioning splitting rules, which divide a selected cluster into two clusters at each level of the hierarchy, are especially popular in Guenoche et al. (1991). Divisive clustering is more computationally intensive than agglomerative clustering, since if
there are $n$ objects, then the first step has $2^{n-1} - 1$ ways to progress forward. Divisive clustering is rarely used in practice.

Dendrograms

The standard graphical device that shows the result of a hierarchical clustering is the dendrogram or tree, which is illustrated in Figure 1.7. The interpretation of the dendrograms for agglomerative and divisive clustering is generally the same. Each node of the dendrogram represents a cluster and its sub-nodes are the subclusters. The height of the nodes represents the distance of two subclusters from one another. Given a cut value of $h$, say $h_0$, as shown in Figure 1.7, there would be three clusters. In the application to microarray gene expression data, the dendrogram is usually combined with a heatmap to provide visualization of profile patterns for both genes and samples. Note that the dendrogram trees are non-unique for a given clustering result. Swapping the locations of the two subnodes horizontally for any node...
does not change the dendrogram. So in a standard dendrogram the relative horizontal distance means little or nothing.

### 1.2.3 Partitioning

Partitioning methods split the data into a specified number of clusters. A partition can be formed by using a hierarchical clustering procedure and cutting the tree at the desired number of clusters. However, such an approach will generally not produce an optimal solution for a particular partitioning problem. The general idea behind most partitioning methods is to cluster objects so that the sum of squared dissimilarities between each object and its closest center from a list of representative centers is minimized. The popular methods in this category are $k$-means and $k$-medoids.

**$k$-means**

$k$-means clustering is based on the idea that there are $k$ centroids which are the centers of the clusters. The centroid is defined as the point in the cluster that minimizes the sum of squared distances between the objects. In other words, it is the ‘center’ or ‘average’ of the cluster. The standard $k$-means algorithm uses the Euclidean distance. Other distances can be used in a similar algorithm but then the ‘center’ of the cluster is not the mean of the cluster. The basic algorithm as described by MacQueen (MacQueen, 1967) is as follows,

**Algorithm 1.2.1. $k$-means clustering**

1. **Create an initial partition of objects into $k$ clusters.**

2. **Calculate the centroid of each of the $k$ clusters.**

3. (a) For object $i$, calculate its distance to each of the centroids.

   (b) **Allocate object $i$ to the cluster which has the closest centroid.**
(c) If the object is reallocated, recalculate centroids based on new clusters.

4). Repeat step 3) for all objects.

5). Repeat steps 3) and 4) until no reallocation occurs.

The initial partitioning is usually based on randomly assigning objects to the $k$ clusters. It is usually a good practice to repeat the $k$-means clustering algorithm using different initial random partitions. Similarly to the hierarchical clustering, there is no guarantee that the resulting clustering structure from the $k$-means algorithm is ‘optimal’ in a statistical sense. More discussion can be found in Hartigan and Wong (1979).

$k$-medoids

$k$-Medoids is analogous to $k$ means, but instead of defining clusters centroids using the average of the objects within the clusters, medoids are found to define each cluster. This is similar to using the median vs. the mean in the univariate setting. One attractive feature of the $k$-medoids approach is that it is more robust against outliers than a $k$-means approach. The algorithm of partitioning around medoids (PAM) proposed by Kaufman and Rousseeuw (2005) is:

Algorithm 1.2.2. $k$-medoids

1). Create an initial $k$ medoids, which are at $k$ data points.

2). (a) For object $i$, calculate its distance to each of the medoids; an examples of the distances is Euclidean distance.

(b) Allocate object $i$ to the cluster with the closest medoid.

\textsuperscript{9}Medoid is an entity $i^{*}$ of a cluster $S$, $i^{*} \in S$, such that it minimizes the sum of distances to other elements of $S$, that is $\sum_{j \in S} d(i^{*}, j) = \min \sum_{j \in S} d(i, j)$. The distance $d(i, j)$ can be any dissimilarity function, such as squared Euclidean distance or Mahalanobis distance.
(c) If the object is reallocated, search or recompute for the new medoids based on the new clusters.

3). Repeat step 2) for all objects.

4). Repeat steps 2) and 3) until no reallocation occur

Note that a centroid in the $k$-means algorithm does not have to be an original data point, but medoid in the $k$-medoids algorithm has to be.

One drawback of the $k$-medoids algorithm is that it is sensitive to the inclusion of unrelated features and the overall noise of the data.

1.2.4 Other Clustering Algorithms

Model-based Clustering

Model-based clustering is a partitioning method in which a probability framework is assumed for the clusters (Everitt and Hand, 1981; Titterington et al., 1985). It assumes that data come from a source of several, say $K$, subpopulations. Each subpopulation has its own probability distribution and the overall population can be a a finite mixture of $K$ subpopulations:

$$f(X) = \sum_{k=1}^{K} p_k f_k(X)$$

where $p_k$’s are the mixing proportions and $f_k(\cdot)$’s denote the density functions of the subpopulations. Normal model-based clustering has been applied to microarray data by McLachlan et al. (2002), Pan et al. (2002), and Yeung et al. (2001a). One advantage of model-based clustering is that one has recourse to a probabilistic framework that can be used to compare across competing clustering results using some statistical model selection criteria such as AIC and BIC. Normal model-based clustering will find spherical or elliptical clusters, but will not find nonconvex structures.
Ward’s clustering

Ward’s clustering refers to the cluster analysis that optimizes Ward’s criterion or Ward’s (Ward, 1963) statistic, which defines the distance between two clusters as the sum of squared distances between clusters divided by the total sum of squares. It can be applied to both hierarchical and partitioning algorithms.

Two-way clustering

The goal of this type of analysis is to identify groups of features associated with some characteristics that are only represented by a subset of samples. This approach is relatively new. Its recent development includes block clustering (Hartigan, 1972; Tibshirani et al., 1999; Alon et al., 1999; Getz et al., 2000), gene shaving (Hastie et al., 2000), and the plaid modeling (Lazzeroni and Owen, 2002).

Currently, a typical microarray experiment contains $10^4 \sim 10^5$ genes, expected to reach $10^6$. However, the number of samples involved is generally in the order of 100. In microarray data mining, clustering on genes or samples is both meaningful. On one hand, co-expressed genes can be grouped into clusters based on their expression patterns (Ben-Dor et al., 1999; Eisen et al., 1998), in which the genes are treated as the objects while the samples are the features. On the other hand, samples can be clustered into groups based on their gene profiles, in which samples are the objects and genes are now the features. This is why two-way clustering is starting to gain popularity for analyzing microarray gene expression data. A subset of genes might define an interesting cluster on samples which might be hard to see if only one-way clustering is applied. Currently in molecular biology, it’s believed that only a small subset of genes participates in any cellular process of interest. The cellular process takes place only in a subset of samples.
Fuzzy clustering

So far we only assume that each object belongs to only one group. The basic idea of the fuzzy clustering method is that objects are allowed to belong to all clusters with different degrees of likelihood (Dunn, 1974; Bezdek, 1981).

There are many new clustering methods. Clustering is a very active research area in data mining and statistics.

1.3 Feature Selection

1.3.1 Introduction

Feature selection, or variable selection, has been an active research area in pattern recognition, statistics, and data mining communities. The main idea of feature selection is to choose a subset of features (variables) by eliminating features with little or no discrimination or prediction power.

There are many potential benefits of feature selection:

- facilitating data visualization and data understanding,
- reducing measurement and storage requirements,
- reducing training time for prediction model building, and
- defining the curse of dimensionality to improve prediction performance.

In statistical model selection literature, the most popular form of feature/variable selection is stepwise regression. It is a greedy algorithm that adds the best feature/variable (or deletes the worst feature/variable) at each step based on some criteria. Examples are variable selection based on Mallow’s Cp statistics, Akaike information criterion (AIC), and Bayesian information criterion (BIC). Efficient algorithms have also been developed for some specific feature structures, such as Branch and Bound, and Piecewise Linear Networks algorithms.
In high-dimensional data analysis, a feature-by-feature test approach is often used to select the most significant features based on testing each individual hypothesis for each feature. The p-values of each of hypotheses are computed, and a stopping criterion based on adjusted p-values is used to control the multiplicity issue in simultaneous testing, such as Bonferroni or FDR adjustment. This kind of test-based microarray analysis can be considered as a feature selection problem if its goal is to find the most significant genes of interest.

Recently, Dy and Brodley (2000), Devaney and Ram (1997), and Agrawal et al. (1998) have combined feature selection and clustering procedures together with a single or unified criterion. For feature selection in an unsupervised learning, learning algorithms are designed to find a natural grouping of the objects in the feature space. Thus feature selection in unsupervised learning aims to find a good subset of features that forms high quality clusters for a given number of clusters. A cluster of high quality is a cluster with highly ‘homogeneous’ data points.

Many variable selection algorithms include variable ranking as a principal or auxiliary selection mechanism because of its simplicity, scalability, and good empirical success. Bekkerman et al. (2003), Caruana and Sa (2003), Forman (2003), and Weston et al. (2003) used variable ranking as a baseline method. Variable ranking is not necessarily used to build predictors of a statistical model.

1.3.2 Feature Selection in Microarray Data Analysis

Microarray experiments enable biologists to monitor expression levels of thousands of genes. Such a dataset poses special challenges for data analysis. The main obstacle is that the number of samples is small due to practical and financial concerns. This results in the situation where the number of features (or genes) well outnumbers the number of observations. The terms
of “curse of dimensionality” and “peaking phenomenon” have been coined in the machine learning and pattern recognition community. They refer to the phenomenon that inclusion of excessive features may actually degrade the performance of classification or clustering if the number of training samples used to build the classifier is relatively small compared to the number of features (Cho et al., 1998). The typical treatment is to reduce the dimensionality of the feature space before classification or clustering using feature extraction and feature selection. Feature extraction algorithms create new features based on transformation and/or combination of original features, while feature selection algorithms aim at selecting a subset of original features. Techniques like principal component analysis (PCA) and singular value decomposition (SVD) have been used to create salient features (Hastie et al., 2000; Khan et al., 2001) for sample classification on microarray datasets. Feature selection, or in our case, gene selection, generates a small set of informative genes, which not only leads to better classification or clustering procedures, but also enables further biological investigation. The variable ranking is one of the methods used commonly in microarray analyses. For example, in drug discovery experiments, a ranking criterion is used to find genes that discriminate between healthy and diseased patients. Such genes may code for “drugable” proteins, or proteins that may themselves be used as drugs.

In order to find the optimal subset of features that maximizes some feature selection criterion function, straightforward implementation would require evaluation of the criterion function for each feature subset, which is a classical NP-hard problem. Various heuristics and greedy algorithms have been proposed to find sub-optimal solutions. Assuming independence between features in single gene-based selection algorithms, discriminative scores for individual genes are calculated, and top ranked genes are combined as a selected gene set. Various discriminative scores have been proposed, including statis-
tical test statistics (t-test, F-test) (Bo and Jonassen, 2002), non-parametric measure like Total Number of Misclassifications (TNoM) (Ben-Dor et al., 2000), mutual information (Xing et al., 2001; Yu and Liu, 2004), S2N ratio (signal to noise ratio) (Golub et al., 1999), extreme value distribution (Grosse and I., 2003), and Significance Analysis of Microarrays (SAM) (Tusher et al., 2001). Although simple, this class of algorithms is widely used in microarray data analysis and has been proven to be effective and efficient.

However, the assumption of independence between genes over-simplifies the complex relationship between genes. Genes are well known to interact with each other through gene regulative networks. As a matter of fact, the common assumption of cluster analysis on a microarray dataset (Jiang and Zhang, 2004) is that co-regulated genes have similar expression profiles. Bo and Jonassen (2002) proposed to calculate discriminant scores for a pair of genes instead of each individual gene. Several recent works on feature selection, especially gene selection (Jaeger et al., 2003; Wu and Zhang, 2004; Yu and Liu, 2004), took into consideration the correlation between genes explicitly by limiting redundancy in the resulting gene set. Heuristically, selected genes need to first have high discriminating scores and secondly have low/no correlation with genes that have already been selected. Generic feature selection algorithms, such as SFFS (sequential forward floating selection) and SBFS (sequential backward floating selection), have also been used for selecting informative genes from microarray datasets.

1.4 Road Map of the Following Chapters

Clustering and feature selection are two commonly applied techniques for high dimensional data such as microarray data. In this thesis, we propose a new clustering and visualization technique, PfCluster, and a new feature selection algorithm, RPselection algorithm, for gene profile data mining.
In Chapter 2, our new profile based clustering method, PfCluster, is presented. A coherence index is proposed to determine proper cutoff values of a dendrogram. The null distribution of the coherence index is studied in detail for the parallel case. Simulation experiments are conducted to evaluate our approach. Applications to two real datasets are given.

In chapter 3, our feature selection algorithm, RPselection algorithm, is developed. The RPselection algorithm uses a resampling-based partitioning to rank genes based on their relevance to a known external sample class. A FIT measure is developed to select the most relevant genes that significantly stand out from others. A simulation study and a real data example are both presented.

In Chapter 4, some usage and details of our software packages, pfclust and rpselect, are shown. pfclust and rpselect are R packages developed in R object-orientated programming framework. They correspond to our PfCluster method and RPselection algorithm, respectively.

In Chapter 5, the connection between the PfCluster method and the RPselection algorithm are illustrated. Some conclusions are drawn and future research is discussed.
Chapter 2

PfCluster: A New Profile Cluster Analysis Procedure

2.1 Introduction

Cluster analysis has been widely applied to many areas of genetics and genomics studies since Eisen et al. (1998). Examples include Perou et al. (1999), Sorlie et al. (2003), Pollack et al. (2002), Ross et al. (2000), Scherf et al. (2000), and Walker et al. (1999). It is now well known that clustering is a powerful method for organizing genes and experimental samples into meaningful groups, based on the similarity measure of gene expression profiles.

Although we will focus on the study of gene expression profiles, the method we propose in this chapter is applicable to other types of data profiles or patterns. A gene expression profile is defined to be a series of measurements recorded under various conditions. The conditions here are broadly defined, for example, as tissue types, cell lines, time points, subjects, treatments, experimental conditions, etc. A profile cluster analysis can be used, for example, to cluster genes into a smaller number of sets so that they can be used to classify tissue samples, to discover genes that belong to the same molecular pathway (Segal et al., 2003a), or to find clusters of genes that are
potentially co-regulated and thus can be used to search for common motifs in upstream regions of the genes in each cluster (Segal et al., 2003b).

The method we propose is called PfCluster. It is motivated by Dr. Jiong Yang’s presentation given in the EECS department at Case Western Reserve University in 2001. This work has been published in Yu et al. (2003). As an example demonstrated in their work, in Figure 2.1 we give profiles of three objects over five conditions, let $\mu_1$, $\mu_2$ and $\mu_3$ denote those three profiles, where $\mu_i = (\mu_{i,b}, \mu_{i,c}, \ldots, \mu_{i,e})$, $i = 1, 2$ and 3. Yu et al. (2003) in their work used a measure called $\max(pScore)$ as the distance measure between two profiles, at only two “time” points. For example, for objects 1 and 2 at conditions $h$ and $j$, $pScore = |(\mu_{h,1} - \mu_{h,2}) - (\mu_{j,1} - \mu_{j,2})|$. In PfCluster, our distance measures are based on the distance at all “time” points. A key step in developing a reasonable measure for all “time” points is finding a suitable contrast matrix and applying the contrast matrix to the vector of pairwise distances.

PfCluster is different from the existing clustering algorithms in other as-
2.2 Profile Patterns

Genes interact with each other. Gene profiles across different experimental conditions may show patterns that reveal the interactions of genes under different conditions. Some of the biologically meaningful gene profile patterns are shown in Figure 2.2, representing parallel, anti-reflective, fan-shaped and shifted profiles between two genes. For example, parallel profiles happen when two genes are co-expressed; anti-reflective profiles occur when two genes functionally suppress each other; shifted profiles with respect to a fixed profile
indicate that some genes follow a leader gene (the fixed profile) in a lagged time; and fan-shaped profiles with respect to a center profile constitute a group or cluster that works around the center gene. It is often convenient to first group parallel profiles into clusters and then study the anti-reflective, fan-shaped or shifted relationship between profiles based on the mean profiles of the parallel clusters.

2.2.1 Parallel Case

Let $\mu_1$ and $\mu_2$ denote two population mean profiles (Fig 2.3), where $\mu_i = (\mu_{i,1}, \mu_{i,2}, \ldots, \mu_{i,t}), i = 1, 2$. In Fig 2.3, $t = 5$, which represents $\mu_1$ and $\mu_2$ across five treatment conditions.

![Figure 2.3: Parallel profile with $t = 5$](image)

To check if $\mu_1$ is parallel to $\mu_2$, we introduce a $(t - 1) \times t$ contrast matrix,
\[ C = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix} \] (2.1)

and a comparison vector,

\[ b = C(\mu_2 - \mu_1)', \] (2.2)

with length

\[ d = \sqrt{b'b}. \] (2.3)

Then \( d^2 = 0 \) iff \( \mu_1 \) is parallel to \( \mu_2 \). This \( d^2 \) can serve as a similarity metric between \( \mu_1 \) and \( \mu_2 \). The smaller \( d \) is, the closer (more parallel) \( \mu_1 \) and \( \mu_2 \) are to each other.

### 2.2.2 Other Cases

For all other profile patterns, similar to the parallel case, the key for building an appropriate similarity metric is to find an appropriate contrast metric and comparison vectors \( \mu_1 \) and \( \mu_2 \). The comparison vector for anti-reflective, fan-shaped and shifted profile patterns can be defined as

\[ b_{ar} = C(\mu_2 + \mu_1)', \quad \text{Anti-reflective} \] (2.4)

\[ b_{fs} = C(\mu_2 - \alpha \mu_1)', \quad \text{Fan-shaped} \] (2.5)

\[ b_s = C(\mu_2(t) - \mu_1(t + \tau))', \quad \text{Shifted} \] (2.6)

where \( C \) is the same as in (2.1), \( \alpha \) and \( \tau \) are two coefficients that can be either known or unknown. We call \( \alpha \) a scale parameter for the fan-shaped case and \( \tau \) a shift parameter for the shifted case. We use the notation \( \mu(t) \) to represent the profile \( \mu \) at time/condition \( t \), and use \( \mu(t + \tau) \) to represent
the profile \( \boldsymbol{\mu}(t) \) with a time/condition lag of \( \tau \) units. We can rewrite all of these \( b \) in a general format as

\[
b = C(\mu_2(t) - \alpha \mu_1(t + \tau))', \tag{2.7}
\]

which can represent all formulations in equations (2.2) and (2.4) - (2.6). Specifically, \( \alpha = 0 \) and \( \tau = 0 \) in (2.2); \( \alpha = -1 \) and \( \tau = 0 \) in (2.4); \( \tau = 0 \) and \( \alpha > 0 \) in (2.5); and \( \alpha = 1 \) in (2.6).

### 2.3 Dissimilarity Matrix

Suppose that we have a data matrix \( X = (x_{ij}) \) for \( i = 1, 2, \ldots, p \) and \( j = 1, 2, \ldots, n \), with \( p \) rows and \( n \) columns representing \( p \) genes and \( n \) samples. If there are no repeated measurements at each of the experimental conditions then \( n \) samples correspond to \( n \) conditions. If there are repetitions, we assume that \( x_{ij} \) is the corresponding average for gene \( i \) under condition \( t \). For gene \( i \) and \( i' \), we let

\[
s_{ii'} = X_{i'}(t) - \alpha X_i(t + \tau), \quad i \neq i', \tag{2.8}
\]

where \( X_i(t) \) corresponds to the \( i \)th row of the data matrix \( X \), and let

\[
b_{ii'} = Cs_{ii'}, \tag{2.9}
\]

where \( C \) is the contrast matrix defined in (2.1). We define a dissimilarity measure between two profiles as

\[
D_{ii'}(\alpha, \tau) = \sqrt{b_{ii'}^T(C\Sigma_aC')^{-1}b_{ii'}}, \tag{2.10}
\]

where \( \Sigma_a \) is the covariance matrix of \( (s_{ii'}) \), which will be estimated if necessary. If \( \alpha \) and \( \tau \) are unknown then

\[
D_{ii'} = \min_{\alpha, \tau} D_{ii'}(\alpha, \tau) \tag{2.11}
\]
and the dissimilarity matrix is thereby defined as \( D = (D_{ii'}) \). The definition of \( D \) is very general; it represents all the profile patterns described in Section 2.2. Furthermore \( D \) for the parallel case has the following properties:

- Non-negativity: \( D_{ii'} \geq 0 \)
- Symmetry: \( D_{ii'} = D_{i'i} \)
- Semi-definiteness: \( D_{ii'} = 0 \) if \( i \) and \( i' \) coincide.

### 2.4 Coherence Index

For each gene we assume for now that its sample profile (perhaps after a transformation) has a multivariate normal distribution. Consider the null condition that \( E(s_{ii'}) = \delta_{ii'} \) is a constant vector of length \( n \),

\[
\delta'_{ii'} = (\Delta_{ii'}, \Delta_{ii'}, \ldots, \Delta_{ii'})_{1 \times n}.
\]  

This null condition represents two parallel profiles if \( \alpha = 1, \tau = 0 \), anti-reflective profiles if \( \alpha = -1, \tau = 0 \), fan-shaped profiles if \( \tau = 0, \alpha > 0 \), and shifted profiles if \( \alpha = 1 \). Therefore, under the null condition (2.12), the \( n \)-dimensional random vector \( s_{ii'} \) in equation (2.8) follows a multivariate normal distribution:

\[
s_{ii'} = X_{i'}(t) - \alpha X_i(t + \tau) \sim N_n(\delta_{ii'}, \Sigma_s),
\]

where \( \Sigma_s \) is the \( n \times n \) dimensional common covariance matrix of \( s_{ii'} \). Thus,

\[
b_{ii'} = Cs_{ii'} \sim N_{n-1}(0, C\Sigma_sC').
\]

Therefore,

\[
D_{ii'}^2 = b_{ii'}'(C\Sigma_sC')^{-1}b_{ii'} \sim \chi^2_{n-1}.
\]

if \( \alpha, \tau, \Sigma_s \) are known or fixed. In the following section, we will develop our coherence indices based on \( D_{ii'} \). Note that if \( \Sigma_s \) is unknown, \( \Sigma_s \) in \( D_{ii'} \) will be
replaced by a good estimate of $\Sigma_s$. If $\alpha, \tau$ are unknown, then $D_{ii'}$ is replaced by
\begin{equation}
\tilde{D}_{ii'} = \min_{\alpha, \tau} D_{ii'}.
\end{equation}

In this thesis, $\alpha$ and $\tau$ are taken to be known in developing the null distributions of our coherence indices below. In the parallel and anti-reflective cases, $\alpha$ and $\tau$ are known. In the fan-shaped and shifted cases, $\alpha$ and $\tau$ may be unknown and may have to be estimated. In this case the null distribution of the coherence index is different from that when $\alpha$ and $\tau$ are known, although the basic PfCluster algorithm is the same.

**Definition 2.4.1. (Coherence index of a cluster).** Let $G$ be a cluster of $p$ profiles, the coherence index $I$ of $G$ is
\begin{equation}
I = P(\max_{i,i' \in G} D_{ii'}^2 > d_G),
\end{equation}
where $d_G$ is the observed value of $D_G = \max_{i,i' \in G} D_{ii'}^2$.

**Definition 2.4.2.** A coherent cluster at level $\alpha$ is a cluster whose coherence index is larger than $\alpha$. So, the coherent clusters are:
\begin{equation}
\{G : I_G > \alpha\}
\end{equation}

**Definition 2.4.3.** A critical value $d^*$ at level $\alpha$ for a coherent cluster is the upper $\alpha$ quantile of $D_G$ under the null condition, s.t.
\begin{equation}
P(D_G > d^*) = \alpha.
\end{equation}

### 2.5 Threshold for Coherence Index

For simplicity, we consider the approximation to the threshold $d^*$ under the parallel case. The threshold $d^*$ under the anti-reflective case is similar and
is omitted. We also assume for now that $\Sigma_s$ is fixed. Recall that $D_G = \max_{i,i' \in G} D_{ii'}^2$. The threshold $d^*$ at level $\alpha$ is the solution $d$ to the equation:

$$P(D_G > d) = \alpha. \quad (2.20)$$

The inclusion-exclusion identity leads to the decomposition

$$P(\max_{i,i' \in G} (D_{ii'}^2) > d) = P(D_{ii'}^2 > d, \text{ for some } i, i' \in G)$$

$$= \sum_{i<i'} P(D_{ii'}^2 > d)$$

$$- \sum \sum_{i<j', j<j'} P(D_{ii'}^2 > d, D_{jj'}^2 > d)$$

$$+ \cdots$$

$$+ (-1)^{n-1} P(D_{ii'}^2 > d \text{ for all } (i, i'), i < i'), \quad (2.21)$$

which will be used to develop approximations to $d^*$ under the null condition (2.12).

### 2.5.1 The First Order Approximation: $d_1^*$

Considering the first term in (2.21) and setting it equal to $\alpha$, we have

$$\sum_{i<i'} P(D_{ii'}^2 > d) = \alpha. \quad (2.22)$$

We call the solution to (2.22) our first order approximation and denote it by $d_1^*$. Since $D_{ii'}^2$’s are identically distributed for all pairs $(i, i')$, where $i \neq i'$, and $D_{ii'}^2 \sim \chi^2_{n-1}$. It is obvious that the left-hand side of (2.22) is $\left(\frac{p}{2}\right) P(D_{12}^2 > d)$. Therefore (2.22) can be rewritten as

$$P(D_{12}^2 > d) = \frac{\alpha}{\left(\frac{p}{2}\right)}$$
and

\[ d_1^* = \chi_{n-1}^2 \left( \frac{2\alpha}{p(p-1)} \right), \tag{2.23} \]

where \( \chi_{n-1}^2 \left( \frac{2\alpha}{p(p-1)} \right) \) denotes the upper \( \frac{2\alpha}{p(p-1)} \) quantile of a Chi-square distribution with \((n-1)\) d.f. Here we assume that there are \( p \) genes and pairwise distances \( D_{ii'}^2 \) are calculated between each pair of those \( p \) genes.

### 2.5.2 The Second Order Approximation: \( d_2^* \)

Considering the 1st two terms in (2.21) and setting it equal to \( \alpha \), we then have

\[
\sum_{i<i'} P(D_{ii'}^2 > d) - \sum_{i<i',j<j'} P(D_{ii'}^2 > d, D_{jj'}^2 > d) = \alpha \tag{2.24}
\]

We call the solution to (2.24) our second order approximation and denote it as \( d_2^* \). It is easily seen that there are two groups, \( G_1 \) and \( G_2 \), in each of which \( P(D_{ii'}^2 > d, D_{jj'}^2 > d) \) has the same value:

- \( G_1 \): \((i, i', j, j')\) are all different. In this case \( D_{ii'}^2 \) and \( D_{jj'}^2 \) are independent to each other approximately. For example,

\[
D_{12}^2 = (C(X_1 - X_2))^T \left(C \Sigma_s C^T\right)^{-1} C(X_1 - X_2)
\]

and

\[
D_{34}^2 = (C(X_3 - X_4))^T \left(C \Sigma_s C^T\right)^{-1} C(X_3 - X_4).
\]

Let \( N_1 \) be the size of \( G_1 \), then

\[
N_1 = \frac{1}{2} \left(\begin{array}{c} p \\ 4 \end{array}\right) = \frac{p(p-1)(p-2)(p-3)}{8}. \tag{2.25}
\]

In this case each \( P(D_{ii'}^2 > d, D_{jj'}^2 > d) \) can be rewritten as

\[
P(D_{ii'}^2 > d, D_{jj'}^2 > d) = (P(D_{ii'}^2 > d))^2 = (P(D_{12}^2 > d))^2.
\]
• \( G_2 \): one of \( i, i' \) is same as one of \( j, j' \). In this case, \( D_{ii'}^2 \) and \( D_{jj'}^2 \) are dependent with one common \( X \) shared by \( D_{ii'}^2 \) and \( D_{jj'}^2 \). For example,

\[
D_{12}^2 = \left( C(X_1 - X_2) \right)^T \left( C\Sigma_s C^T \right)^{-1} C(X_1 - X_2)
\]

and

\[
D_{13}^2 = \left( C(X_1 - X_3) \right)^T \left( C\Sigma_s C^T \right)^{-1} C(X_1 - X_3).
\]

Let \( N_2 \) be the size of \( G_2 \) where

\[
N_2 = \binom{p}{3} \binom{3}{1} = \frac{1}{2} p(p - 1)(p - 2). \tag{2.26}
\]

Let \( d_2^* \) to be the solution of \( d \) to the equation:

\[
N_0 P(D_{12}^2 > d) - N_1 (P(D_{12}^2 > d))^2 - N_2 P(D_{12}^2 > d, D_{13}^2 > d) = \alpha, \tag{2.27}
\]

where

\[
N_0 = \binom{p}{2} = \frac{p(p - 1)}{2}. \tag{2.28}
\]

In order to solve (2.27) we need to derive \( P(D_{12}^2 > d, D_{13}^2 > d) \), which is shown below.

For a positive definite covariance matrix \( \Sigma_s \), we can apply Cholesky decomposition to get a non-singular \( A \) such that,

\[
AA^T = C\Sigma_s C^T. \tag{2.29}
\]

Let

\[
Z_{12} = A^{-1} C(X_2 - X_1),
\]

\[
Z_{13} = A^{-1} C(X_3 - X_1), \tag{2.30}
\]

and denote \( z_{2l} \) and \( z_{3l} \) as the elements of \( Z_{12} \) and \( Z_{13} \), where \( l = 1, 2, \ldots, n - 1 \).

Since \( X_1 \), \( X_2 \) and \( X_3 \) are from an \( iid \) \( n \)-dimensional normal distribution,
\(N(\mu_x, \Sigma_x)\), then under the null condition (2.12),

\[
E(Z_{12}) = A^{-1}CE(X_2 - X_1) = 0,
\]

\[
E(Z_{13}) = A^{-1}CE(X_3 - X_1) = 0,
\]

\[
Var(Z_{12}) = Var(A^{-1}C(X_2 - X_1))
\]

\[
= A^{-1}Var(C(X_2 - X_1))A^{-1T}
\]

\[
= A^{-1}C\Sigma_x C^T A^{-1T}
\]

\[
= A^{-1}AA^T A^{-1T}
\]

\[
= I_{(n-1)\times(n-1)},
\]

\[
Var(Z_{13}) = I_{(n-1)\times(n-1)}.
\]

Therefore,

\[
z_{2l}, z_{3l} \sim N(0, 1), \ l = 1, 2, \ldots, n-1.
\]

Now we derive the correlation between \(Z_{12}\) and \(Z_{13}\),

\[
Corr(Z_{12}, Z_{13}) = \frac{Cov\left(A^{-1}C(X_2 - X_1), A^{-1}C(X_3 - X_1)\right)}{\sqrt{Var\left(A^{-1}C(X_2 - X_1)\right) \cdot Var\left(A^{-1}C(X_3 - X_1)\right)}}.
\]

where,

\[
Cov\left(A^{-1}C(X_2 - X_1), A^{-1}C(X_3 - X_1)\right)
\]

\[
= A^{-1}C\left(Cov\left(X_2 - X_1, X_3 - X_1\right)\right)C^T A^{-1T}
\]

\[
= A^{-1}C\Sigma_x C^T A^{-1T}
\]

and

\[
Var\left(A^{-1}C(X_2 - X_1)\right) = Var\left(A^{-1}C(X_3 - X_1)\right) = 2A^{-1}C\Sigma_x C^T A^{-1T}.
\]
Therefore,
\[
Corr(Z_{12}, Z_{13}) = \frac{1}{2} A^{-1} C \Sigma_x C^T A^{-1} \left( A^{-1} C \Sigma_x C^T A^{-1} \right)^{-1}
\]
\[
= \frac{1}{2} I_{(n-1) \times (n-1)}.
\]

Denoting the correlation between \(z_{2l}\) and \(z_{3l}\) by \(\rho\), we have
\[
\rho = Corr(z_{2l}, z_{3l}) = \frac{1}{2}, \quad l = 1, 2, \ldots, n - 1,
\]
and \(Corr(z_{2l}, z_{3l'}) = 0\), if \(i = i'\).

The conditional distribution of \(z_{2l} | z_{3l}\) is,
\[
z_{2l} | z_{3l} = N(\rho z_{3l}, 1 - \rho^2), \quad l = 1, 2, \ldots, n - 1. \quad (2.31)
\]

Hence it is easily seen that,
\[
\frac{1}{(1 - \rho^2)} \sum_{l=1}^{n-1} z_{2l}^2 | (z_{3l}, \quad l = 1, 2, \ldots, n - 1) \sim \text{noncentral } \chi^2(\lambda), \quad (2.32)
\]
where \(\kappa\) and \(\lambda\) are the d.f. and the noncentrality parameter, respectively, so
\[
\kappa = n - 1,
\]
\[
\lambda = \frac{\rho^2}{1 - \rho^2} \sum_{l=1}^{n-1} z_{3l}^2. \quad (2.33)
\]

Since the noncentral Chi-square distribution can be computed from a Poisson mixture of central Chi-square distributions with \(\kappa + 2t\) d.f., where \(t\) is a Poisson r.v. with parameter \(\lambda/2\) (Johnson et al., 1994),
\[
P\left( \sum_{l=1}^{n-1} z_{2l}^2 \leq d \left| \sum_{l=1}^{n-1} z_{3l}^2 \right. \right) = \sum_{j=0}^{\infty} \frac{e^{-\frac{\lambda}{2}} \left(\frac{\lambda}{2}\right)^j}{j!} P_{Y_{\kappa+2j}} \left( Y_{\kappa+2j} \leq \frac{d}{1 - \rho^2} \right), \quad (2.34)
\]
where \(\sum_{l=1}^{n-1} z_{3l}^2 \sim \chi_{n-1}\) and \(Y_{\nu}\) denotes a r.v. that is distributed as Chi-square with \(\nu\) d.f.. We use \(f_{Y_{\nu}}(\cdot)\) to denote the pdf of \(Y_{\nu}\), then
\[
f_{Y_{\nu}}(y) = \frac{1}{2^{\frac{\nu}{2}} \Gamma\left(\frac{\nu}{2}\right)} y^{\frac{\nu}{2}-1} e^{-\frac{y}{2}}. \quad (2.35)
\]
Let us now consider \( P(D_{12}^2 > d, D_{13}^2 > d) \). Recall the transformation in (2.30). We have

\[
P(D_{12}^2 > d, D_{13}^2 > d) = P\left((C(X_2 - X_1))^T(C\Sigma_s C)^{-1}C(X_2 - X_1) > d, (C(X_3 - X_1))^T(C\Sigma_s C)^{-1}C(X_3 - X_1) > d\right)
\]

\[
= P\left(Z_{12}^T Z_{12} > d, Z_{13}^T Z_{13} > d\right)
\]

\[
= P\left(\sum_{l=1}^{n-1} z_{2l}^2 > d, \sum_{l=1}^{n-1} z_{3l}^2 > d\right)
\]

\[
= \int_d^{\infty} P\left(\sum_{l=1}^{n-1} z_{2l}^2 > d | \sum_{l=1}^{n-1} z_{3l}^2 = y\right) \frac{1}{2^{\frac{n-1}{2}} \Gamma\left(\frac{n-1}{2}\right)} y^\frac{n-3}{2} e^{-\frac{y}{2}} dy
\]

\[
= \int_d^{\infty} \left(1 - \sum_{j=0}^{\infty} \frac{\lambda^\frac{j}{2}}{j!} P(Y_{n-1+2j} \leq \frac{d}{1 - \rho^2})\right) \frac{1}{2^{\frac{n-1}{2}} \Gamma\left(\frac{n-1}{2}\right)} y^\frac{n-3}{2} e^{-\frac{y}{2}} dy
\]

\[
= P(\chi^2_{n-1} \geq d) - \sum_{j=0}^{\infty} \frac{1}{j!} P(Y_{n-1+2j} \leq \frac{d}{1 - \rho^2}) \int_d^{\infty} \frac{e^{-\lambda \frac{y}{2}}}{2^{\frac{n-1}{2}} \Gamma\left(\frac{n-1}{2}\right)} y^\frac{n-3}{2} e^{-\frac{y}{2}} dy
\]

\[
= P(\chi^2_{n-1} \geq d) - \sum_{j=0}^{\infty} \frac{1}{2^{\frac{n-1}{2}} \Gamma\left(\frac{n-1}{2}\right) j!} P(Y_{n-1+2j} \leq \frac{d}{1 - \rho^2}) \int_d^{\infty} e^{-\lambda \frac{y}{2}} y^\frac{n-3}{2} e^{-\frac{y}{2}} dy.
\]

(2.36)
Substituting \( \lambda \) in (2.33), the term \( \int_0^d e^{-\frac{1}{2}(\frac{\lambda}{2})y} f_{Y_{n-1}}(y) dy \) above can be rewritten as

\[
\int_d^\infty e^{-\frac{1}{2}(\frac{\lambda}{2})y} \frac{n-1}{2} e^{-\frac{y}{2}} dy
\]

\[
= \int_d^\infty e^{-\frac{\rho^2 y}{2(1-\rho^2)}} \frac{j}{2} y \frac{n-1}{2} e^{-\frac{y}{2}} dy
\]

\[
= \int_d^\infty \left( \frac{\rho^2 y}{2(1-\rho^2)} \right)^j y \frac{n-1}{2} e^{-\frac{y}{2}} dy
\]

\[
= \int_d^\infty \left( \frac{\rho^2 y}{(1-\rho^2)} \right)^j \frac{1}{2} y \frac{n-1}{2} e^{-\frac{y}{2}} dy
\]

\[
= (1-\rho^2)^{\frac{n-1}{2}} \left( \frac{\rho^2}{2} \right)^j \int_d^\infty \left( \frac{y}{1-\rho^2} \right)^{\frac{n-1}{2}+j-1} e^{-\frac{y}{2(1-\rho^2)}} d\left( \frac{y}{1-\rho^2} \right)
\]

\[
= (1-\rho^2)^{\frac{n-1}{2}} \left( \frac{\rho^2}{2} \right)^j \int_d^\infty t^{\frac{n-1}{2}+j-1} e^{-\frac{t}{2}} dt, \quad \text{with } t = \frac{y}{1-\rho^2}
\]

\[
= \frac{\Gamma\left(\frac{n}{2}+j\right)(1-\rho^2)^{\frac{n-1}{2}} \rho^{2j}}{\Gamma\left(\frac{n-1}{2}\right)} P(Y_{n-1+2j} > \frac{d}{1-\rho^2}).
\]

Therefore,

\[
P\left(D_{12}^2 > d, D_{13}^2 > d\right)
\]

\[
= P\left(\lambda_{n-1}^2 \geq d\right)
\]

\[
- \sum_{j=0}^{\infty} \frac{1}{j!} P(Y_{n-1+2j} \leq \frac{d}{1-\rho^2}) \frac{\Gamma\left(\frac{n-1}{2}+j\right)(1-\rho^2)^{\frac{n-1}{2}} \rho^{2j}}{\Gamma\left(\frac{n-1}{2}\right)} P(Y_{n-1+2j} > \frac{d}{1-\rho^2})
\]
\[
= P(\chi^2_{n-1} \geq d) \\
- \sum_{j=0}^{\infty} \frac{1}{j!} (1 - P(Y_{n-1+2j} > \frac{d}{1 - \rho^2})) \frac{\Gamma(\frac{n-1}{2} + j)(1 - \rho^2)^{\frac{n-1}{2} + j} \rho^{2j}}{\Gamma(\frac{n-1}{2})} P(Y_{n-1+2j} > \frac{d}{1 - \rho^2}) \\
= P(\chi^2_{n-1} \geq d) - \sum_{j=0}^{\infty} \frac{\Gamma(\frac{n-1}{2} + j)(1 - \rho^2)^{\frac{n-1}{2} + j} \rho^{2j}}{\Gamma(\frac{n-1}{2})j!} P(\chi^2_{n-1+2j} > \frac{d}{1 - \rho^2})^2 \\
+ \sum_{j=0}^{\infty} \frac{\Gamma(\frac{n-1}{2} + j)(1 - \rho^2)^{\frac{n-1}{2}} \rho^{2j}}{\Gamma(\frac{n-1}{2})j!} \left( P(\chi^2_{n-1+2j} > \frac{d}{1 - \rho^2}) \right)^2 \\
= P(\chi^2_{n-1} \geq d) - \sum_{j=0}^{\infty} B_j P(\chi^2_{n-1+2j} > \frac{d}{1 - \rho^2}) + \sum_{j=0}^{\infty} B_j \left( P(\chi^2_{n-1+2j} > \frac{d}{1 - \rho^2}) \right)^2 \\
\] (2.38)

where

\[
B_j = \frac{\Gamma(\frac{n-1}{2} + j)(1 - \rho^2)^{\frac{n-1}{2} + j} \rho^{2j}}{\Gamma(\frac{n-1}{2})j!}. \\
\] (2.39)

**Simulation study of \( d_1^* \) and \( d_2^* \)**

Let \( X \sim MVN_n(\mu, \Sigma_x) \) be a data matrix for \( p \) genes on \( n \) conditions. For the parallel case in (2.7), we can simulate the distribution of \( \max(D_{ii^*}^2) \) with the following steps,

1). Define a population mean profile \( \mu \).

2). Generate data, \( X^{(1)}, X^{(2)}, \ldots, X^{(p)} \), independently from \( MVN_n(\mu, \Sigma_x) \).

3). Get an estimate of \( \Sigma_s \) following the steps in Section 2.6.

4). Compute \( D_{ii^*}^2 \) using (2.8), (2.9) and (2.10).

5). Obtain the \( \max_{i,i^* \in G}(D_{ii^*}^2) \) of \( D_{ii^*}^2 \)’s computed in step 4).

6). Repeat steps 1) - 5) \( B \) times (\( B=10,000 \)) to get a simulated distribution for \( \max(D_{ii^*}^2) \), denoted as \( \hat{F}_{\max(D_{ii^*}^2)} \).
For a given set of $\alpha$ we can estimate $d^*$ from $\hat{F}_{\max(D_{\alpha}^2)}$, where $d^*$ is the upper $\alpha$ quantile of $\hat{F}_{\max(D_{\alpha}^2)}$. The first order approximation to $d^*$, $d_1^*$, is obtained by evaluating (2.23). The second order approximation, $d_2^*$, is calculated by solving the equation (2.24). No closed-form solution is available for the second order approximation and so a numerical result is pursued.

In Figures 2.4 and 2.5, true $d^*$ (simulated) are compared with its first and second order approximations $d_1^*$ and $d_2^*$, for $n = 5$ and 10, $p = 10, 20, 50, 100, 200, 500$, and $\alpha = 0.01, \cdots, 0.1$. Here $n$ is the number of condition points, $p$ is the number of genes in a cluster. As shown in Figures 2.4 and 2.5, $d_1^*$ is closer to $d^*$ for small $p$ and small $\alpha$. $d_2^*$ is more accurate than $d_1^*$ also for small $p$ and $\alpha$. $d_2^*$ may be numerically unattainable for large $p$ or $\alpha$. Overall, both $d_1^*$ and $d_2^*$ have a consistent pattern to that of $d^*$, although $d_1^*$ tends to be conservative and $d_2^*$ tends to be liberal. Thus, either of them can still be used to guide us in a certain degree to decide if a cluster is coherent. For example, if a cluster is coherent at the level $\alpha$ based on $d_1^*$, it must be coherent at level $\alpha$ based on $d^*$. If a cluster is incoherent at the level $\alpha$ based on $d_2^*$, it must be incoherent at level $\alpha$ based on $d^*$. It is also possible to perform a regression analysis to derive an estimate of $d^*$ based on $d_1^*$ or $d_2^*$ using the simulated results for $p = 10, 20, \cdots, 2000$. We will show the applications of using $d_1^*$ in §2.9 for simulated data and in §2.10 for two real data examples. Our coherence index works remarkably well.

In Chapter 5, we will further discuss this simulation results, provide an insight into why we get such behavior on $d_1^*$ and $d_2^*$, and suggest future research.
Figure 2.4: Comparison of $d^*$, $d_1^*$, and $d_2^*$, $s^2 = 0.01$: 1st-order approximation $d_1^*$, 2nd-order approximation $d_2^*$ and the exact $d^*$ based on the simulated data. The data is simulated using a known diagonal covariance matrix with all the diagonal elements, $s^2 = 0.01$. 
Figure 2.5: Comparison of $d^*$, $d_1^*$, and $d_2^*$, $s^2 = 0.05$: 1st-order approximation $d_1^*$, 2nd-order approximation $d_2^*$ and the exact $d^*$ based on the simulated data. The data is simulated using a known diagonal covariance matrix with all the diagonal elements, $s^2 = 0.05$. 
2.6 Estimation of $\Sigma_s$

2.6.1 Estimation of $\Sigma_x$

In (2.8), for iid $X_i$ and $X_i'$ with a common variance $\Sigma_x$, $\Sigma_s$ can be written as

$$\Sigma_s = Var(s_{ii'}) = (1 + \alpha^2)\Sigma_x.$$ 

For the parallel case where $\alpha = 1$, we have,

$$\Sigma_s = 2\Sigma_x. \quad (2.40)$$

Therefore, we can first estimate $\Sigma_x$ and then estimate $\Sigma_s$ using the above relation.

For a $p \times n$ data matrix $X$, the simple estimate of $\Sigma_x$ of $X^T$ is

$$\hat{\Sigma}_x = \frac{1}{P-1} \sum_{i=1}^{P} (X_i - \bar{X})(X_i - \bar{X})^T, \quad (2.41)$$

where

$$X_i = (x_{i1}, x_{i2}, \cdots, x_{in})^T,$$

$$\bar{X} = \frac{1}{P} \sum_{i=1}^{P} X_i.$$ 

An improved estimate of $\Sigma_x$ when data are clustered is to use the pooled variance estimate

$$\hat{\Sigma}_x = \frac{\sum_{k=1}^{K}(n_k - 1)\Sigma_{x,k}}{\sum_{k=1}^{K}(n_k - 1)}, \quad k = 1, 2, \cdots, K, \quad (2.42)$$

where $K$ is the number of clusters, $n_k, k = 1, 2, \cdots, K$ are the cluster sizes, and $\Sigma_{x,k}, k = 1, 2, \cdots, K$ are their respective $n \times n$ covariance matrices. Next we use an adaptive procedure to estimate $\Sigma_x$ based on the idea above:
1). Cluster data $X$ using $D_{it}^2$ with the simple variance-covariance estimate as described in (2.41).

2). Choose a cut value $h$ from the dendrogram obtained in 1) and get data clusters.

3). Calculate the pooled variance-covariance estimate $\hat{\Sigma}_x$ using (2.42).

**2.6.2 Dendrogram Cut-value $h$ for Estimating $\Sigma_x$**

The strategy here is to find the best cut-value of the dendrogram height $h$ (see Section 1.2.2) so that the resulting clusters are most homogeneous, and the pooled variance of those homogeneous clusters can be used as the final variance estimate of $X$. Therefore, we propose to find the cut-value $h$ of the PfCluster dendrogram such that the following measure $M$ is minimized,

$$M = \sum_{j=1}^{n} \left( \nu \log s_j^2 - \sum_{k=1}^{K} \nu_k \log(s_{j,k}^2) \right)^2,$$

where

$$\nu = \sum_{k=1}^{K} \nu_k, \nu_k = n_k - 1,$$

$$s_j^2 = \frac{\sum_{k=1}^{K} \nu_k s_{j,k}^2}{\nu}, s_{j,k}^2 = \frac{1}{n_k - 1} \sum_{i=1}^{n_k} (x_{ij} - \bar{X}_{j,k})^2,$$

$$\bar{X}_{j,k} = \frac{1}{n_k} \sum_{i=1}^{n_k} x_{ij}.$$

The size and the variance of cluster $k$ are $n_k$ and $s_{j,k}^2$, where $s_j^2$ is the pooled variance from all the clusters $k = 1, \cdots, K$. If the variance of data is stabilized, we expect a common variance across different clusters when the dendrogram is cut at a proper height $h$. This assumption of homogeneity is
reasonable for microarray data, because data variance is usually stabilized in data preprocessing. The core part inside $\sum_{j=1}^{n}$ in (2.43) has the same form as the likelihood formulation used in Bartlett’s test of homogeneity and the $\sum_{j=1}^{n}$ is taken over all dimensions of $X$.

**Remark.** If $\Sigma_s$ is estimated, the Chi-square distribution of $D_{ii'}^2$ is an $F$ distribution which is approximately Chi-square. Therefore, these $d^*, d_1^*, d_2^*$ are the approximated critical values.

### 2.7 Distribution of $D_{ii'}^2$ for Positive Semidefinite $\Sigma_s$

In order for (2.15) to be valid, $C\Sigma_sC'$ in (2.14) needs to be positive definite. When $C\Sigma_sC'$ is positive semidefinite, we will revise the distribution of $D_{ii'}^2$ based on the following theorem.

**Theorem 2.7.1.** (Schott, 2005) Let $b \sim N_n(0, \Sigma)$, where $\Sigma$ is positive semidefinite, and suppose that $A$ is an $n \times n$ symmetric matrix, if (a): $\Sigma A\Sigma A \Sigma = \Sigma A\Sigma$ and (b): $tr(A\Sigma) = r$, then $b'Ab \sim \chi_r^2$.

Thus if we can find a proper $A$ for $C\Sigma_sC'$, then $D_{ii'}^2$ in (2.15) will still have a Chi-squared distribution but with $d.f.r < n - 1$.

**Remark.** A matrix $A$ satisfying condition (a) and (b) in Theorem 2.7.1 is $\Sigma^+$, the Moore-Penrose inverse of $\Sigma$. So if $b \sim N_n(0, \Sigma)$ then $b'\Sigma^+b \sim \chi_r^2$, where $r = rank(\Sigma)$.

Therefore a robust version of (2.15) when $C\Sigma_sC'$ is positive semidefinite is

$$D_{ii'}^+ = \sqrt{b_{ii'}'(C\Sigma_sC')^+b_{ii'}},$$

(2.44)

where $(C\Sigma_sC')^+$ is the Moore-Penrose inverse of $C\Sigma_sC'$. 
From Theorem 2.7.1 and the remark above we have
\[ D_{ii'}^2 \sim \chi_r^2, \quad (2.45) \]
where \( r = \text{rank}(C\Sigma_sC') \).

## 2.8 PfCluster

What set apart of our PfCluster from the existing clustering algorithm is that a coherent index is introduced first time. Base on such a coherence index, the quality of resulting clusters can be assessed by using the coherence index as an internal quality measure, and questions, such as, which cluster is coherent and how many clusters are coherent can be answered based on user-provided coherence criteria.

In addition, the PfCluster allows for a class of new distance metrics for profile pattern discovery. Those profile-based distances provide better biological relevance than the commonly used correlation coefficient based distance or the standardized Euclidean distance.

We have discussed the PfCluster dissimilarity measure, the coherence index, and the mechanism of choosing a coherence index threshold for coherent clusters. In the rest of the section, we will focus on other key components in PfCluster as a tool to be used for real data analysis: data preprocessing, choice of a clustering algorithm, and graphical representation.

### 2.8.1 Preprocessing

Data preprocessing includes data quantitation, data transformation, standardization or normalization, data filtering, and feature selection or extraction.

Data quantitation involves identification of spots with real signals from other artifacts. For microarray data, this is the process of image analysis described in Chapter 1.
Data transformation, standardization or normalization

For data that are measured under different units, standardization is needed to make data features (columns) comparable so that all columns contribute to data scatter equally. This is commonly done by shifting and rescaling the data. Let $X$ denote the original data and $Y$ denote the transformed data of $X$, let $X_j$ and $Y_j$ be the $j$th rows of data $X$ and $Y$, respectively. As an example, for $X_j$, the standardization is obtained by applying

$$Y_j = \frac{(X_j - a)}{b}. \quad (2.46)$$

The shift coefficient $a$ can be $0$, $\text{min}$, $\text{max}$, $\text{midrange}$, $\text{grand mean}$, or $\text{median}$ of $X_j$, and by default PfCluster uses $\text{mean}$ as $a$. The scale coefficient $b$ could be $\text{range}$, $\text{standard deviation}$, or $\text{grand mean}$ of $X_j$, and PfCluster by default uses $\text{standard deviation}$. We recommend to consider the $\text{range}$, not the $\text{standard deviation}$, when data are obviously multi-modal. In case there are possible outliers, which might highly affect the range, we could use the difference between the 90th and 10th percentiles instead of the range (Mirkin, 2005; Milligan and Cooper, 1988).

Transformation is done for different purposes. For example, in microarray data, a log or generalized log transformation can be performed to stabilize the variance.

For preprocessing of microarray data, because of the image nature of microarray raw data, and the batch processing of microarray experiments, background correction and array-wise normalization are needed.

Data filtering

**Simple filtering: Most interesting profile filtering**

In practice it may be necessary to quickly select a subset from a huge data set to make the data computationally manageable. Such a subset can be chosen
in many ways. Here we provide a simple filtering algorithm called *most interesting profile (MIP) filtering*, which can efficiently select the genes with interesting profiles. Let’s again assume we have the data matrix $X = (x_{ij})$ with $p$ rows and $n$ columns. Each gene profile is represented by a row vector, $X_i = (x_{i1}, x_{i2}, \ldots, x_{in}), i = 1, 2, \ldots, p$. For each profile, say $X_i$, we define a statistic $T_i$ to measure how far away each profile $X_i$ is from its center which is an uninteresting flat profile, $(\bar{x}_i, \bar{x}_i, \ldots, \bar{x}_i)$.

$$T_i = \max \{|x_{i1} - \bar{x}_i|, |x_{i2} - \bar{x}_i|, \ldots, |x_{in} - \bar{x}_i|\},$$

(2.47)

where $\bar{x}_i = \frac{1}{n} \sum_{j=1}^{n} x_{ij}$. Then we look at the distribution of $\log(T_i)$ and pick up those genes with large $\log(T_i)$ values. In our MIP filtering implemented in PfCluster, by default genes with top 10% of $\log(T_i)$ values are chosen.

**CV filtering**

Another effective filtering is based on the coefficient of variation of all the data points and hence is called CV filtering. For a gene profile $X_i = (x_{i1}, x_{i2}, \ldots, x_{in}), i = 1, 2, \ldots, p$, we define,

$$V_i = \frac{S_i}{\bar{x}_i},$$

(2.48)

where

$$S_i = \sqrt{\frac{1}{n} \sum_{j=1}^{n} (x_{ij} - \bar{x}_i)},$$

$$\bar{x}_i = \frac{1}{n} \sum_{j=1}^{n} x_{ij}.$$

(2.49)

Since we are only interested in the profiles that are variable and informative, we should focus only on the profiles with large $V_i$. CV filtering in PfCluster by default uses $V_i > 0.1$. 
**Design-based filtering**

Another commonly used filtering scheme is based on an appropriate statistic chosen according to the underlying experimental design of a data set. For example, if we are interested in a comparative profile analysis over a two-group design, such as for data obtained from two treatment groups, two types of samples, or two experimental conditions, we can filter the genes based on a *t* test or a Mann-Whitney-Wilcoxon test. In this case, we calculate the test statistic for each of the genes and select the genes with large test statistics or small p-values.

**Feature selection/extraction**

Profiles are generally defined as measurements recorded or calculated across different samples or conditions. Sometimes raw data are already in a convenient form for profile analyses. For example, profiles from longitudinal data measured over time, or from experimental-specific data measured over several conditions are already in a convenient form. In some other times, feature extraction is needed to first find a subspace of data that retains the most important information and then profiles are constructed over this subspace. A good illustrative example is in protein profiling. In protein profiling we need to extract a small number of peaks from the protein spectra and then transform the spectra into profiles over those selected peaks before performing a profile cluster analysis. It is also possible that we need to create some ‘synthetic’ features (variables) that contain the most important information of a dataset. For example, profiles can be constructed using the derived covariant information from a regression analysis.

In many other cases, we might need to perform a feature selection due to too many features in the data. Data examples in this category include microarray data and network traffic data.
Therefore, feature selection or extraction here refers to subsetting or transforming a dataset into a convenient profile dataset without losing its important information before a further profile data analysis.

In chapter 3 we will propose a feature selection algorithm based on a resampling-based partitioning (RPselection algorithm). For microarray data, the RPselection algorithm can select genes that are most relevant to a clinical outcome. A profile cluster analysis can then be performed as a downstream analysis.

2.8.2 Clustering Algorithm

Hierarchical clustering

In our PfCluster package, the default clustering algorithm is the agglomerative hierarchical clustering method as described in Everitt et al. (2001). The default similarity metric is

\[ D_{ii'}(\alpha, \tau) = \sqrt{b_{ii'}(C_{i}C_{i}')^{-1}b_{ii'}} \]

with \( \alpha = 0 \) and \( \tau = 0 \) for the case of parallel profiles. With the agglomerative clustering approach a hierarchy is established in a bottom-up fashion, by starting from single-member ‘clusters’, sequentially merging them into ‘parental’ nodes and finally reaching the root node that contains all the members.

Agglomerative clustering algorithms differ when different within- and between-cluster distances are used. The three common choices of between-cluster distance are those by single, complete and average linkages, in which the between-cluster distances are \( \min_{i \in C_1, j \in C_2} d_{ij} \), \( \max_{i \in C_1, j \in C_2} d_{ij} \) and \( \text{ave}_{i \in C_1, j \in C_2} d_{ij} \) respectively, where \( d_{ij} \) is the distance between the \( i \)th element in cluster \( C_1 \) and the \( j \)th element in cluster \( C_2 \). Lance and Williams (1967) also introduced a general recurrence formula to flexibly define the distance between
two groups by giving a set of parameters. Quite a broad set of agglomerative algorithms can be defined using the Lance and Williams formula. This formula can represent all interesting algorithms proposed in the literature so far (Table 4.2 in Everitt et al. (2001)), with an appropriate choice of parameters.

Function \textit{hclust} (Murtagh, 1985) and function \textit{agnes} in package \textit{cluster} (Kaufman and Rousseeuw, 2005) are the available agglomerative hierarchical algorithms in R. Function \textit{hclust} includes Ward’s minimum variance method; complete linkage, single linkage, and average linkage methods; centroid, median and mcquitty methods (MacQuitty, 1966). Function \textit{agnes} includes average linkage, complete linkage, single linkage, and weighted average linkage methods, as well as a flexible method that uses the Lance Williams formula to specify the dissimilarities. The default clustering algorithm in PfCluster is \textit{hclust} with the complete linkage.
Visualization

The dendrogram is a mathematical and pictorial representation of a tree structure. It has been commonly used to display the results from an agglomerative clustering. In Figure 2.6 we illustrate the terminology used in the dendrogram. There are three types of nodes: root node, internal node and terminal nodes. The root node represents the largest cluster that contains all the members, internal nodes represent the intermediate-level clusters in the hierarchy and terminal nodes represent single-member clusters. The height or the length of stem represents the distance at which each fusion is made. Topology of dendrogram indicates the arrangement of nodes and stems in the tree. The names of objects (e.g. a, b,... in Fig 2.6 and Fig 2.7) attached to the terminal nodes are known as labels.

Given a dataset of size n and a clustering procedure, there are $2^{n-1}$ different possible arrangements of a dendrogram. The resulting dendrograms can be compared using cophenetic correlation, and Goodman and Kruskal’s
\( \gamma \) as described in Everitt et al. (2001).

Dendrograms from PfCluster extend the regular dendrograms by including at each of the selected heights coherence index value as a measure of internal cluster quality. In the agglomerative clustering, at each height where a new cluster forms, the coherence index \( I \) in (2.17) is calculated for that cluster. Hence, if a user provides a threshold for the coherence index, say \( \alpha \), then any coherent clusters based on \( \alpha \) are indicated in the PfCluster dendrogram, and those coherent clusters are also annotated in the PfCluster dendrogram by their indices.

A PfCluster dendrogram is given in Fig 2.7, where PfCluster is applied to 10 most interesting genes, filtered out using the MIP filtering (Section 2.8.1) from the folate acid data described in Section 2.10.2 below. Branches labeled 1, 2 and 3 are the coherent clusters with coherence index threshold \( \alpha \) chosen to be 0.6. Coherent cluster 1 contains \( a \) and \( b \) as its members with the coherence index \( I = 0.714 \), coherent cluster 2 contains member \( d \) and \( e \) with \( I = 0.975 \), and coherent cluster 3 contains members \( f, g \) and \( h \) with \( I = 0.769 \).

2.9 Simulation Studies

2.9.1 Study I

In this study there are 5 profile groups, each of which has a distinct population mean profile, denoted as, \( \mu_i, i = 1, \ldots, 5 \). Each mean profile, \( \mu_i \), has 5
Figure 2.8: Profile patterns of the simulated data I

time or condition points. Specifically, the mean profiles are

\[ \mu_1 = (-1 \ 1 \ -1 \ 1 \ -1)' \]
\[ \mu_2 = (1 \ -1 \ 1 \ -1 \ 1)' \]
\[ \mu_3 = (0 \ 1 \ 2 \ 1 \ 0)' \]
\[ \mu_4 = (0 \ -1 \ -2 \ -1 \ 0)' \]
\[ \mu_5 = (0 \ 0 \ 0 \ 0 \ 0)' , \] (2.50)

where \( \mu_5 \) represents a flat profile with no change across the 5 conditions. See Figure 2.8 for a graphical representation of these 5 mean profiles.

For each of those profile groups, represented by \( \mu_i \), we generate \( n_i \) profiles based on,

\[ X_{i1}, \ldots, X_{im_i} \sim MVN(\mu_i, \Sigma_i) \] (2.51)

where \( MVN(\mu_i, \Sigma_i) \) is a 5-dimensional multinormal distribution with mean
The number of genes in each of 5 groups are 20, 10, 80, 20 and 1000, respectively. So, the majority of profiles are in the uninteresting flat profile group with mean profile $\mu_5$. The simulated data I is shown in Figure 2.9 and Figure 2.10

We applied PfCluster to the simulated data I with a coherence index cutoff value $\alpha = 0.05$. Its dendrogram is shown in Figure 2.11, the detected profile groups are presented in Figure 2.12, and the corresponding heatmap is provided in Figure 2.13. The heatmap in Figure 2.13 is different from the one in Figure 2.10, because the genes are now ordered based on the dendrogram.

**Figure 2.9: Data profiles of the simulated data I**

$\mu_i$ and covariance matrix $\Sigma_i$:

$$
\Sigma_i = \begin{pmatrix}
0.05 & 0 & 0 & 0 & 0 \\
0 & 0.05 & 0 & 0 & 0 \\
0 & 0 & 0.05 & 0 & 0 \\
0 & 0 & 0 & 0.05 & 0 \\
0 & 0 & 0 & 0 & 0.05 \\
\end{pmatrix}.
$$

(2.52)
Figure 2.10: Heatmap of the simulated data I

Figure 2.11: PfCluster dendrogram of the simulated data I
Figure 2.12: Profile groups of the simulated data I recovered by PfCluster
Figure 2.13: Heatmap of the simulated data I with genes ordered by the PfCluster dendrogram
produced by PfCluster. The profile groups discovered by PfCluster match 100% with the true ones we used to generate the simulated data I.
Similarly to Study I, we simulate another data set, called the simulated data II, also with 5 distinct profile groups but with a more continuous mean curve, measured at 20 different conditions. The population mean profiles of those 5 profile groups are:

\[ \mu_1 = (\mu_{1,1}, \mu_{1,2}, \ldots, \mu_{1,20})', \]  
\[ \mu_2 = (\mu_{2,1}, \mu_{2,2}, \ldots, \mu_{2,20})', \]  
\[ \mu_3 = (\mu_{3,1}, \mu_{3,2}, \ldots, \mu_{3,20})', \]  
\[ \mu_4 = (\mu_{4,1}, \mu_{4,2}, \ldots, \mu_{4,20})', \]  
\[ \mu_5 = (\mu_{5,1}, \mu_{5,2}, \ldots, \mu_{5,20})', \]  

where \( \mu_5 \) also represents the flat profile. Similarly, for each of 5 profile groups, say \( \mu_i \), we generate \( n_i \) data profiles based on a multivariate normal
are given in Figure 2.19. The profile groups discovered by PfCluster also match 100% the true data profiles of the simulated data II. The dendrogram is presented in Figure (2.54).

\[
\Sigma_i = \begin{pmatrix}
0.01 & 0 & 0 & 0 & 0 \\
0 & 0.01 & 0 & 0 & 0 \\
0 & 0 & 0.01 & 0 & 0 \\
0 & 0 & 0 & 0.01 & 0 \\
0 & 0 & 0 & 0 & 0.01 \\
\end{pmatrix}
\]

The number of genes, \(n_i, i = 1, 2, \ldots, 5\), in 5 profile groups is 20, 100, 80, 20 and 1000, respectively, with the majority of profiles having the uninteresting flat mean profile \(\mu_5\). The pattern of the simulated data II is shown in Figure 2.15 and Figure 2.16.

We apply PfCluster to the simulated data II with a coherence index cutoff value \(\alpha = 0.05\). Its dendrogram is presented in Figure 2.17, uncovered groups are given in Figure 2.18, and the corresponding heatmap is shown in Figure 2.19. The profile groups discovered by PfCluster also match 100% the true profile groups specified.
Figure 2.16: Heatmap of the simulated data II

Figure 2.17: PfCluster dendrogram of the simulated data II
Figure 2.18: Profile groups of the simulated data II recovered by PfCluster
Figure 2.19: Heatmap of the simulated data II with genes ordered by the PfCluster dendrogram
2.10 Real Data Applications

In this section, we apply PfCluster to two real datasets to explore their profile patterns.

2.10.1 Biomarker LOH data

The objective of this study is to improve patient management and identify novel compartments to target therapy at the etiologic level. A total of 122 formalin-fixed, paraffin-embedded, primary head and neck squamous cell carcinomas (SCC) from 122 patients have been analyzed for genomic instability, using 366 microsatellite markers in the neoplastic epithelium and its surrounding stroma compartment. These samples were consecutively selected for squamous cell histology that are not being known to have received previous chemo-radiotherapy in proximity to the resection and that are not within a clinical trial. Genome-wide loss of heterozygosity or allelic imbalance scan (LOH/AI) is conducted over all markers and their LOH frequencies are recorded. Additionally, the data of the clinical stage of those 122 patients are known. Here the interest is to discover and group the LOH profiles that vary over clinical stages, to give some insight into the downstream data analysis.

The profiles here are defined to be the LOH frequencies of markers measured at the 4 clinical stages. All the epithelial-specific LOH profiles (Stroma-specific LOH are not shown) and their corresponding heatmap are shown in Figure 2.20 and Figure 2.21, respectively.

We apply PfCluster to the LOH data with a coherence index cutoff threshold $\alpha = 0.05$, with the results shown in Figures 2.22,2.23 and 2.24. The corresponding heatmap in Figure 2.26 is more organized than the one in Figure 2.21 because markers are ordered based on the dendrogram produced from PfCluster. There are total of 38 different profile groups discovered by PfCluster with different profile patterns as shown in Figures 2.23 and 2.24.
Figure 2.20: Data profiles of LOH data

Figure 2.21: Heatmap of LOH data
Figure 2.22: *PfCluster* Dendrogram of LOH data
Figure 2.23: Profile groups 1-24 of LOH data by PfCluster
Figure 2.24: Profile groups 25-38 of LOH data by PfCluster
One of the study goals is to identify LOH at loci (biomarkers) that are positively associated with the aggressiveness of disease, as reflected by a linear trend of LOH over the clinical stages 1-4. The coherent profile groups in Figures 2.23 and 2.24 give us insight for downstream analyses to identify those loci. For example, the coherent cluster 5 containing a group of markers that show clear positive association with the clinical stages. The further investigation on this group shows that the LOH of ‘D6S305’ (locus 6q26) occurred significantly more frequently in clinical stages 3 and 4 than in stages 1 and 2. In our profile plot in Figure 2.25 (Group 5 in Figure 2.23), we see a linear increase of LOH frequencies from stage 1 (13%) and stage 2 (22%) to stage 3 (35%) and to stage 4 (50%) for locus ‘6q26’, which contains the common fragile site\textsuperscript{1} FRA6E.

\textsuperscript{1}Fragile sites are loci or regions that are especially sensitive to forming gaps or breaks on metaphase chromosomes when DNA replication is perturbed. Fragile sites, and the genes that lie within fragile site regions, are frequently deleted or rearranged in many cancer cells.
Figure 2.26: Heatmap of LOH data with markers ordered by the PfCluster dendrogram
2.10.2 Folate Acid Data

The deficiency of folate acid can cause birth defect, heart disease, colon cancer and other cancers. In order to study the folate acid pathway and regulatory network on human, gene expression from liver samples, together with corresponding folate acid levels and homocysteine values have been collected by Dr. Toshimori Kitami in the Department of Genetics at Case Western Reserve University. The design of the study is shown in Fig 2.27. There are seven treatments in the experiment: FOL1, FOL2, FOL7, FOL14, FOL15, FOL21 and CTL0, and each treatment is a combination of folate depleted diet and control diet supplied to mice in a given period of time. For example in FOL1, mice were supplied with a control diet for first 7 days and then with folate depleted diet for the next day, and in FOL21, mice were supplied with the control diet for the first 7 days and with folate depleted diet for the next 14 days and the control diet for the last 7 days. For each treatment, 8 replicate female mice were used. They were pooled into one sample and measured with 2-color cDNA microarray using a referenced design as shown in Fig 2.28. Each sample was measured twice with the dye labeling swapped. Two strains of mice were measured, indicated as A and B. In the demo example below, strain A samples with one of the reference design were used to show how PfCluster works.

In this case there are a total of 13,468 genes. The data used was the ratio of intensities between the sample and the reference. The gene expression profiles were constructed using 7 treatments as sample points, in the order of CTL0, FOL1, FOL7, FOL14, FOL15 and FOL21. So each profile gives the gene expression ratio for a specific gene varying over the seven treatment conditions, in that specific order.

We applied PfCluster to the data, which are first standardized with shift parameter $a$ using mean and scale parameter $b$ using standard deviation, and
Figure 2.27: Experimental Design. For each treatment (for example FOL1), there were 8 replicate female mice. The 8 replicates were pooled into one sample and measured twice with microarray. For each time point, there are two measurements. Each strain of mice (A or B) followed the same experimental design above.

Figure 2.28: Microarray reference design of the folate acid data
Figure 2.29: Gene expression profiles of top 0.1% of genes in Folate data
Figure 2.30: PfCluster dendrogram using the top 0.1% of genes in Folate data
Figure 2.31: Resulting clusters, Group 1 - 6, from PfCluster using top 0.1% of genes in Folate data.
Figure 2.32: Resulting clusters, Group 7 - 12, from PfCluster using top 0.1% of genes in Folate data.
Figure 2.33: Resulting clusters, Group 13 - 1, from PfCluster using top 0.1% of Folate data.
then the top 0.1% of genes are selected using the MIP filtering defined in §2.8.1. The profiles of those genes are shown in Fig 2.29. The PfCluster dendrogram is shown in Fig 2.30. Using $\alpha = 0.05$, 15 coherent clusters are detected. The profiles of all 15 clusters are shown in Fig 2.31, Fig 2.32 and Fig 2.33. Each plot contains the gene profiles from one of the 15 clusters and the middle line in each plot is the average profile based on all the profiles in that cluster. The coherence index $I$ is also listed at the bottom of each plot.

### 2.11 Comparison with Correlation Coefficient Based Approach

In this section, we compare the performance of our PfCluster and the same agglomerative hierarchical clustering algorithm that uses the correlation coefficient based dissimilarity measure. Obviously, one major advantage of our method is that it directly assesses the quality of the resulting clusters based on the coherence index, and provides a number of significantly coherent clusters according to a user-provided $\alpha$.

The dendrograms generated using both PfCluster distance and correlation coefficient based distance for the simulated data I, simulated data II and LOH data are shown in Figure 2.34, Figure 2.35 and Figure 2.36, respectively. It is easily seen by inspecting the tree structure of the dendrograms that, for the simulated data I and II, dendrograms from PfCluster represent the true profile group better. The dendrograms of LOH data are similar based on Pfcluster distance and correlation coefficient.

Another approach to compare the dendrograms is to look at the consistency of the resulting dendrograms with the original distance measure. Hierarchical clustering imposes a hierarchical structure on data. It is usually necessary to consider whether this detected structure is acceptable or introduces unacceptable distortion of the original relationships among the
Figure 2.34: Compare PfCluster (left) with correlation coefficient based clustering (right) using the simulated data I.

Figure 2.35: Compare PfCluster (left) with correlation coefficient based clustering (right) using simulated data II.

Figure 2.36: Compare PfCluster (left) with correlation coefficient based clustering (right) using the LOH data.
objects. A good dissimilarity measure should also be robust on different linkage functions used by the clustering. From this perspective we will compare the dendrogram from the hierarchical cluster analysis with the original distance measure and also compare the dendrograms of the different linkage methods. A good dissimilarity measure should give a better consistency and robustness across different linkage methods. A measure commonly used for comparing a dendrogram both with a proximity matrix and with a second dendrogram is the **cophenetic correlation coefficient**.

### 2.11.1 Cophenetic Correlation

The cophenetic correlation of a cluster dendrogram is defined as the linear correlation coefficient between the cophenetic distances obtained from the dendrogram, and the original distances (or dissimilarities) used to construct the dendrogram. Thus, it is a measure of how faithfully the dendrogram represents the dissimilarities among observations. The **cophenetic distance** between two observations in a dendrogram is represented by the height of the link at which those two observations are first joined. That height is also the distance between the two subclusters that are merged by that link. The cophenetic correlation $c$ is defined as

$$
c = \frac{\sum_{i<j} (d_{ij} - \bar{d})(z_{ij} - \bar{z})}{\sqrt{\sum_{i<j} (d_{ij} - \bar{d})^2 \sum_{i<j} (z_{ij} - \bar{z})^2}}, \quad (2.55)
$$

where,

- $d_{ij}$ is the distance between objects $i$ and $j$.
- $z_{ij}$ is the cophenetic distance between objects $i$ and $j$.
- $\bar{d}$ and $\bar{z}$ are the average of $d_{ij}$ and $z_{ij}$, respectively.
Table 2.1: Compare cophenetic correlation coefficient for PfCluster and correlation coefficient based clustering using the simulated data I

<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>Single</th>
<th>Complete</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single</td>
<td>0.88</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complete</td>
<td>0.82</td>
<td>0.80</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>0.93</td>
<td>0.94</td>
<td>0.85</td>
<td>1</td>
</tr>
</tbody>
</table>

The magnitude of $c$ should be very close to 1 for a high-quality solution. In comparing two dendrograms, $d_{ij}$ and $z_{ij}$ are the cophenetic distance between objects $i$ and $j$ from the first dendrogram and the second dendrogram, respectively.

Agglomerative hierarchical clustering is applied on the simulated data I, II, and LOH data using the PfCluster dissimilarity and correlation coefficient based dissimilarities with different linkage methods, e.g., single linkage, complete linkage and average linkage. Listed in the tables 2.1, 2.2 and 2.3 are the cophenetic coefficients among the original dissimilarity matrix, the dendrograms from the single linkage, complete linkage and average linkage clustering, using both PfCluster and the correlation coefficient based clustering.

For simulated data I and II as indicated by the larger cophenetic coefficients shown in the lower part of the first column in Tables 2.1 and 2.2, single, complete and average linkage methods from PfCluster, all represent the original data much better than the corresponding correlation coefficient based counter parts. Also the larger cophenetic coefficients between each
<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>Single</th>
<th>Complete</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PfCluster</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Original</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single</td>
<td>0.98</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complete</td>
<td>0.94</td>
<td>0.96</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>0.98</td>
<td>1.00</td>
<td>0.96</td>
<td>1</td>
</tr>
<tr>
<td><strong>Correlation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Original</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single</td>
<td>0.37</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Complete</td>
<td>0.41</td>
<td>0.65</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>0.49</td>
<td>0.77</td>
<td>0.68</td>
<td>1</td>
</tr>
</tbody>
</table>

**Table 2.2:** Compare cophenetic correlation coefficient for PfCluster and correlation coefficient based clustering using the simulated data II

<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>Single</th>
<th>Complete</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PfCluster</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Original</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single</td>
<td>0.51</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Complete</td>
<td>0.61</td>
<td>0.38</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>0.68</td>
<td>0.61</td>
<td>0.70</td>
<td>1</td>
</tr>
<tr>
<td><strong>Correlation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Original</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single</td>
<td>0.36</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Complete</td>
<td>0.63</td>
<td>0.38</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>0.73</td>
<td>0.49</td>
<td>0.67</td>
<td>1</td>
</tr>
</tbody>
</table>

**Table 2.3:** Compare cophenetic correlation coefficient for PfCluster and correlation coefficient based clustering using LOH data
pair of single, complete and average linkage methods indicate a much better consistency than the correlation coefficient based clustering in using different linkage methods for the simulated data I and II. Similarly, for LOH data, we can see that the PfCluster performed competitively with the correlation coefficient based approach.

### 2.12 Discussion

Missing values are common in huge data sets and need to be handled, especially in the framework of profile cluster analysis. When the percent of missing values is small and they appear missing at random, one may simply delete them. This is the option we use currently in PfCluster. More sophisticated treatment to missing values need to be developed.

Data standardization is performed sample-wise to make samples (variables or features) comparable. The common principle is that all the samples should contribute equally to the distance measure when no one particular choice of weights on these samples is clearly better than the others. In PfCluster, the contrasts brought in by the contrast matrix in (2.8) place an equal-weight for the measurements across different times or conditions. Therefore, data standardization before the PfClustering is important and is implemented in our PfCluster.

Feature extraction described in Section 2.8 can also be crucial. We propose to use simple feature extraction procedure, such as, MIP, before PfClustering. Our new feature selection procedure RPseletion algorithm in Chapter 3 can be also used for this purpose if necessary. Our RPselection is much more comprehensive than MIP, and can be used as a downstream analysis tool to rank genes.

Coherence index, $I$, as a measure of internal cluster quality, enables PfCluster to assess the quality of resulting clusters and to estimate the number
of clusters in the data. In our software, only the first order approximation $d_1^*$ of $d^*$ are implemented for finding the coherent clusters. We have computed analytically the second order approximation, which provides more accurate approximation to $d^*$ than $d_1^*$ does in certain ranges of $p$ and $\alpha$. In our future research we will investigate higher order approximations to $d^*$ that fit for general $p$ and $\alpha$. We will also explore alternative, computationally efficient approaches to approximate $d^*$.

PfCluster has been configured to use the parallel profile dissimilarity defined in Section 2.3 for mining parallel profile patterns. However, an adaptive approach will be implemented to search through profiles to find the genes that have the anti-reflective, fan-shaped or shifted relationship based on the group profiles resulted from PfCluster.

In PfCluster, data are assumed to be normally distributed after a transformation. This is a reasonable assumption for real microarray data after a logarithm or a generalized logarithm transformation and a suitable normalization.
Chapter 3

RPselection: A New Feature Selection Procedure

3.1 Introduction

One of the basic goals in microarray data analyses is to select genes that are relevant to a sample classification. For example, we might be interested in selecting genes that are responsible for breast cancer, or genes that are responsive to the change of a biological condition, such as, different dose levels of a treatment. Generally we call those disease status (cancer or no cancer) or biological conditions the sample classification. To find those responsible or responsive genes, we look for genes that are differentially expressed among the groups defined by a sample classification. The simplest case is to seek genes that show different transcript abundance between two different groups. More generally, one may search genes whose abundance is associated with multiple factor levels of one or more sample characteristics. Furthermore, one may search genes that are responsive to some continuous-valued sample characteristics. A typical statistical analysis strategy for selecting genes in all of the above situations is to perform a statistical test for each gene (Claverie,
The choice of test depends on the biological interest and the nature of the data. For example, when comparing over two conditions, one might use Student’s \( t \)-test if data are normally distributed, or Wilcoxon rank-sum test if data are not normally distributed, or even a permutation test if one would like to avoid the distributional assumption on data (Dudoit et al., 2002b). In the case when there are continuous-valued sample characteristics, one might want to test for genes that show nonzero coefficients under a regression model, such as a linear model or a Cox proportional hazard model.

There are two problems with the test-based approaches described above. First, there are thousands of genes from a much smaller number of samples in a microarray dataset. So the number of features, \( p \), is very large, in a range from several thousands to tens of thousands, but the number of samples within each condition is relatively much smaller. This is called a large \( p \), small \( n \) problem. Second, genes are correlated but clearly the gene-by-gene test-based approaches do not incorporate the correlation structure among the genes.

A different approach is to select informative genes (Golub 1999; Alizadeh 2000) based on a known sample classification. This approach is a feature selection problem in statistical pattern recognition or machine learning. The goal is to identify the best subset of genes according to the maximal correspondence of a clustering procedure with respect to a target sample classification. A target sample classification can be a known clinical parameter of patients or some phenotype of the samples, to which we will refer as an external class label/source. Under this feature selection framework, we can think of the above test-based approaches as in a special category.

Clustering procedures are effective data mining tools for grouping gene expression profiles or for exploring the patterns in microarray expression data (Alon et al., 1999; Ben-Dor et al., 1999; Eisen et al., 1998). Different dissimilarity metrics can lead to different clustering procedures. Correlation
coefficient has been commonly used as a useful dissimilarity metric for genetic profiling. Our challenges in developing an effective clustering procedure for microarray data analysis include: for a given large number of genes (or features), there are usually multiple clustering structures in the data; different subsets of features could give rise to different cluster structures; and in the large-p-small-n setting, the effectiveness of a small set of informative genes may be masked by a large number of irrelevant genes (called masking genes).

In this chapter, we propose a new feature selection algorithm to find a subset of informative genes that is significantly in favor of an external class source by optimizing a fitness score. Our fitness score measures the relevance between the partition labels from a clustering procedure and the external class label. The method is essentially an ordering and searching algorithm using a partitioning clustering procedure based on the pairwise dissimilarity matrices. This topic is related to supervised learning (since the external class label is available), and also tightly related to the topic of two-way clustering (Getz et al., 2000, 2003) but with a different goal of feature selection using partitioning clustering as an optimization problem. Its strategy is similar to the procedure of comparing two groups by a discriminant analysis, but a typical discriminant analysis procedure generally fails with large p small n genomic data. Our approach is to use partitioning clustering (based on pairwise distances) which will be computationally efficient and naturally accounts for the gene by gene interaction by evaluating every gene in a group that contains others. This is biologically more meaningful than a gene-by-gene procedure because every gene works together with other genes to play their genetic roles in a functional network.
3.2 Methods

Suppose data are collected on \( n \) subjects, \( s_1, s_2, \ldots, s_n \). For each subject, there are \( p \) features, \( x_1, x_2, \ldots, x_p \), measured as a \( p \)-dimensional vector. In the context of microarray data analysis each feature is a gene. Let \( X \) denote the data matrix with \( p \) rows and \( n \) columns corresponding to the observed values of \( p \) features from \( n \) subjects:

\[
X = \begin{pmatrix}
  x_{11} & x_{12} & \cdots & x_{1n} \\
  x_{21} & x_{22} & \cdots & x_{2n} \\
  \vdots & \vdots & \ddots & \vdots \\
  x_{p1} & x_{p2} & \cdots & x_{pn}
\end{pmatrix}_{p \times n}
\]  (3.1)

Let \( \mathcal{X}_d \) denote all possible subsets of size \( d \) from the original \( p \) features, and \( \mathcal{X}_d^{(x_i)} \subset \mathcal{X}_d \) with the additional constraint that the set of the selected \( d \) genes contains the gene \( x_i \).

Consider a partition by a decision rule \( g \) that assigns \( n \) subjects, \( s_1, s_2, \ldots, s_n \), into \( N \) disjoint groups. The \( g \) depends on data \( X \), denoted as \( g(\cdot|X) \); so for each \( s_i \), \( g(s_i|X) \in \{1, \ldots, N \} \). Assume we also have a known class label \( \kappa = (\kappa(s_1), \kappa(s_2), \ldots, \kappa(s_n)) \) from other source of information, which labels the subjects into \( K \) classes, i.e., \( \kappa(s_i) \in \{1, 2, \ldots, K \} \). In this chapter, we assume that \( N = K \), so the number of classes from the partition clustering procedure \( g(\cdot|X) \) is equal to the number of classes in \( \kappa \). However, \( N \) generally can be different from \( K \) because the number of groups presented from a data partition can be different from that in \( \kappa \).

A typical feature selection procedure is defined to be a process that optimizes a criterion function \( J \), which assigns a discrimination performance measure to each feature. Thus the features can be ordered based on the performance measure as

\[
J(x_1) \geq J(x_2) \geq \ldots \geq J(x_p).
\]
Therefore, the $r$ largest $J$ values correspond to the most important $r$ features. Here we propose $J(x)$ to be the average measure of the correspondence between the partition clustering procedure $g(\cdot)$ and the known class label $\kappa$ in partitioning the $n$ subjects into $K$ groups,

$$J(x) = \mathbb{E}\left\{ \phi\left( g(X^{(x)}), \kappa \right) \right\}, \quad (3.2)$$

where $X^{(x)} \in X^{(x)}_d$, $g(X^{(x)})$ denotes $g(\cdot|X^{(x)})$; $\phi$ is a function that evaluates the agreement of the clustering procedure $g(\cdot)$ (perhaps after a permutation) with respect to the known class label $\kappa$; the expectation is taken uniformly over all subsets in $X^{(x)}_d$.

Our rationale of using $J$ in (3.2) as a performance measure is that if $x$ is an important gene (informative gene) then its inclusion in building clusters is beneficial in terms of maximizing on average the agreement between the resulting partition label $g(X^{(x)})$ and the known class label $\kappa$. In the following sections we will propose appropriate $g$, $\phi$ and an estimate of $J$, and describe computational enhancement used in developing our RPselection procedure.

### 3.2.1 Partition Clustering Procedure $g(\cdot)$

We choose to use a $k$-medoids procedure as our partition clustering procedure $g(\cdot)$. The $k$-medoids procedure is similar to a typical $k$-means procedure, introduced in Chapter 1.

**K-medoids**

The aim of a $k$-means algorithm is to partition data into $k$ clusters based on a nearest-neighbor rule so that the within-group sum of squares is minimized. We propose to use the specific Partitioning Around Medoids (pam) algorithm from Kaufman and Rousseeuw (1990). The pam algorithm is more robust than the $k$-means algorithm because it minimizes a sum of more robust dissimilarities than a sum of squared Euclidean distances. Next, we
will specify the dissimilarity metric to be used with pam for gene expression pattern discovery.

**Dissimilarity Metric**

It has been shown that correlation coefficient-based pairwise distance is an effective measure for similar patterns on microarray data when genes are co-regulated or co-expressed (Eisen et al., 1998; Alon et al., 1999; Azuaje, 2003; Ben-Dor et al., 1999; Allocco et al., 2004). Denote measurements on \( p \) features from subjects \( i \) and \( j \) by two vectors, \( (x_{i1}, x_{i2}, \ldots, x_{ip})' \) and \( (x_{j1}, x_{j2}, \ldots, x_{jp})' \). The dissimilarity \( d_{ij} \) based on the Pearson correlation coefficient between subject \( i \) and \( j \) is

\[
d_{ij} = \frac{1 - R_{ij}}{2},
\]

where

\[
R_{ij} = \frac{\sum_{k=1}^{p} (x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)}{\sqrt{\sum_{k=1}^{p} (x_{ik} - \bar{x}_i)^2} \sqrt{\sum_{k=1}^{p} (x_{jk} - \bar{x}_j)^2}};
\]

\[
\bar{x}_i = \frac{1}{p} \sum_{k=1}^{p} x_{ik}, \quad \bar{x}_j = \frac{1}{p} \sum_{k=1}^{p} x_{jk}.
\]

Therefore, \( 0 \leq d_{ij} \leq 1 \). Based on this dissimilarity measure, two subjects are similar if their gene profiles are positively correlated and dissimilar if those profiles are negatively correlated in a linear fashion. Two extreme cases occur when two subjects \( i \) and \( j \) are parallel \( (d_{ij} = 0) \) and are completely opposite to each other \( (d_{ij} = 1) \).

Another option is to use the Spearman rank correlation coefficient. Intuitively, Spearman rank correlation is a measurement of monotone association between \( (x_{i1}, x_{i2}, \ldots, x_{ip})' \) and \( (x_{j1}, x_{j2}, \ldots, x_{jp})' \). It is defined as

\[
R_{ij} = 1 - \frac{6 \sum_{l=1}^{p} \delta_{ijl}^2}{p(p^2 - 1)},
\]

where \( \delta_{ijl}^2 \) is the squared difference between the ranks of \( x_{il} \) and \( x_{jl} \).
where
\[ \delta_{ijl}^2 = (r(x_{il}) - r(x_{jl}))^2, \] (3.5)
in which \( r(x_{il}) \) and \( r(x_{jl}) \) are the ranks of \( x_{il} \) and \( x_{jl} \), respectively. So, \( \delta_{ijl}^2 \) is the difference in statistical ranks of \( x_{il} \) and \( x_{jl} \).

Notice that the above dissimilarities both satisfy the conditions that:
(i) \( d_{ij} \geq 0 \), (ii) \( d_{ii} = 0 \), and (iii) \( d_{ij} = d_{ji} \).

### 3.2.2 Correspondence Measure \( \phi \)

Instead of choosing a criterion to minimize the expected classification error rate, we use a criterion that maximizes the correct classification rate, which is particularly helpful if we have multiple classes, i.e., \( K > 2 \). For each subject \( i \), as before, let \( g(s_i) \) be the class label assigned by the partition clustering procedure \( g(\cdot) \), and \( \kappa(s_i) \) be the class label from the known class label \( \kappa \). Since \( N = K \), \( g(s_i), \kappa(s_i) \in \{1, \ldots, K\}, i = 1, \ldots, n \), we define

\[ \phi_n(g(X), \kappa) = \frac{1}{n} \sum_{i=1}^{n} I(g_0(s_i) = \kappa(s_i)), \] (3.6)

which is the proportion of \( s_i \)'s that \( g(s_i) \) and \( \kappa(s_i) \) agree. Here \( g_0 \) is such that

\[ \frac{1}{n} \sum_{i=1}^{n} I(g_0(s_i) = \kappa(s_i)) = \max_{g_c} \frac{1}{n} \sum_{i=1}^{n} I(g_c(s_i) = \kappa(s_i)), \] (3.7)

where \( g_c \) is a specification of class ids from \( g(\cdot) \). Since any group from the clustering procedure \( g(\cdot) \) can be called group 1, any permutation of the ids does not change the underlying partitions by \( g(\cdot) \). Therefore the maximization in (3.7) is over all permutation of the class ids.

**Estimation of \( \phi_n \)**

The \( \phi_n \) in (3.6) is essentially the correct classification rate of the partition clustering procedure \( g(\cdot) \) with respect to the external known label \( \kappa \). Next
we develop a practical scheme to obtain \( \max_{g_c} \frac{1}{n} \sum_{i=1}^{n} I(g_c = \kappa) \) in (3.7).

Let \( U = (u_{ij}) \) and \( W = (w_{ij}) \) denote the membership matrices, defined by class label \( g_c \) from the partition clustering procedure \( g(\cdot) \), and the external class label \( \kappa \), respectively. Specifically, \( u_{ij} = 1 \) if subject \( s_i \) belongs to class \( j \) and \( u_{ij} = 0 \) otherwise; and \( w_{ij} \) is defined similarly. Thus,

\[
u_{ij} \in \{0, 1\}, w_{ij} \in \{0, 1\},
\]

\[
\sum_{j=1}^{K} u_{ij} = 1, \sum_{j=1}^{K} w_{ij} = 1. \quad (3.8)
\]

An example of \( U \) and \( W \) for \( n \) subjects classified into \( K = 3 \) groups is demonstrated as,

\[
U = \begin{pmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
\vdots & \vdots & \vdots \\
1 & 0 & 0
\end{pmatrix}_{n \times 3}, \quad W = \begin{pmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
\vdots & \vdots & \vdots \\
0 & 1 & 0
\end{pmatrix}_{n \times 3}.
\]

Let \( u_{i}, w_{i}, i = 1, 2, \ldots, n \), be the \( i \)th row of the membership matrices \( U \) and \( W \), respectively. It can easily be seen that a good concordance measure, denoted as \( \phi(U, W) \), between two membership matrices \( U \) and \( W \) is,

\[
\phi(U, W) = \sum_{i=1}^{n} u_{i}'w_{i} = tr(U'W).
\]

So,

\[
\frac{1}{n} \sum_{i=1}^{n} I(g_c(s_i) = \kappa(s_i)) \propto tr(U'W), \quad (3.9)
\]

where \( \propto \) is the “proportional to” sign. Hence

\[
\phi_n (g(X), \kappa) \propto \max_H \left( tr(U'WII) \right), \quad (3.10)
\]
where $\Pi = (\pi_{ij})$ is any permutation matrix, s.t. $\Pi \Pi' = I$, see Appendix A.1; and the max is performed over all permutation matrices $\Pi$ of order $K$.

In order to find $\max_{\Pi} \left( tr(U'W\Pi) \right)$, define $C = (c_{ij})$ to be

$$C = 1 - U'W.$$  \hspace{1cm} (3.11)

Then, ignoring the proportional constant, we can set

$$\phi_n(g(X), \kappa) = \min_{\Pi} \left( tr(C\Pi) \right) = \min_{\Pi} \sum_{i=1}^{K} \sum_{j=1}^{K} c_{ij} \pi_{ij},$$ \hspace{1cm} (3.12)

where the permutation matrix, $\Pi = (\pi_{ij})$, satisfies,

$$\sum_{j=1}^{K} \pi_{ij} = 1, \sum_{i=1}^{K} \pi_{ij} = 1, \pi_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \ldots, K.$$ \hspace{1cm} (3.13)

Now, the estimation of the performance measure $\phi_n$ becomes a Linear Sum Assignment Problem as described in the Appendix A.3.

**Other Estimation of $\phi_n$ that allow $N \neq K$**

The above estimate of $\phi_n$ is for the case that $N = K$. Recall $N$ and $K$, are the numbers of clusters, from a partition clustering procedure $g(\cdot)$, and the external class label $\kappa$, respectively. For more general case that $N \neq K$, we propose to use the Rand Index and Mutual Information to estimate $\phi_n$.

I. **Rand Index**

Given $n$ subjects, there are $\binom{n}{2}$ distinct pairs of subjects. They can be categorized into 3 different types according to the cluster labels from the partition clustering procedure $g(\cdot)$ and the known external class label $\kappa$. Those pairs belong to:

a). the same class by both the partition clustering procedure $g(\cdot)$ and the know label $\kappa$. 
b). a different class by both \( g(\cdot) \) and \( \kappa \).

c). the same class in one of \( g(\cdot) \) and \( \kappa \), but not both.

Let \( A \) be the total number of pairs of a) and b) above. The agreement between \( g(\cdot) \) and \( \kappa \) can be assessed using information contained in an \( N \times K \) table with elements \( n_{ij} \), where \( n_{ij} \) denotes the number of subjects belonging to both the same class by \( g(\cdot) \) in its \( i \)th cluster and by \( \kappa \) in its \( j \)th category.

Define

\[
n_i = \sum_{j=1}^{K} n_{ij}, \quad n_j = \sum_{i=1}^{N} n_{ij}.
\]

It can be shown that

\[
A = \binom{n}{2} + 2 \sum_{i=1}^{N} \sum_{j=1}^{K} \binom{n_{ij}}{2} - \left( \sum_{i=1}^{N} \binom{n_i}{2} + \sum_{j=1}^{K} \binom{n_j}{2} \right).
\]  \(3.14\)

The Rand Index (Rand, 1971) is defined as,

\[
R = \frac{2A}{n(n-1)}.
\]  \(3.15\)

A corrected Rand Index (Hubert and Arabie, 1985) is

\[
R_c = \frac{\sum_{i=1}^{N} \sum_{j=1}^{K} \binom{n_{ij}}{2} - \frac{\sum_{i=1}^{N} \binom{n_i}{2} \sum_{j=1}^{K} \binom{n_j}{2}}{\binom{n}{2}}}{\frac{\sum_{i=1}^{N} \binom{n_i}{2} + \sum_{j=1}^{K} \binom{n_j}{2}}{2} - \frac{\sum_{i=1}^{N} \binom{n_i}{2} \sum_{j=1}^{K} \binom{n_j}{2}}{\binom{n}{2}}}.
\]  \(3.16\)

\( R_c \) is corrected to ensure that its maximum value is 1 and its expected value is 0 when the partitions are selected at random.
II. Mutual Information

In probability and information theory, the mutual information of two random variables is a quantity that measures the mutual dependence of the two variables. Let $Y$ and $Z$ be two random variables. The mutual information between $Y$ and $Z$ is defined in terms of their marginal and joint probability density functions $p(y)$, $p(z)$ and $p(y,z)$ as,

$$I(Y, Z) = \sum_y \sum_z p(y, z) \log \left\{ \frac{p(y, z)}{p(y) \cdot p(z)} \right\}, \quad (3.17)$$

where $\sum$ is over all possible values $y$ and $z$ of $Y$ and $Z$.

Let us now consider that partition clustering procedure $g(\cdot)$ that partitions the subjects into $N$ clusters, and denote each cluster with a number, $h$, where $h \in \{1, 2, \ldots, N\}$. Similarly the external class label $\kappa$ assigns the subjects into $K$ categories. Denote each category as a number, $l$, where $l \in \{1, 2, \ldots, K\}$, here $N \neq K$. Let $n^{(h)}$ be the number of subjects in the cluster $h$ according to $g(\cdot)$, and $n_l$ be the number of subjects in category $l$ according to $\kappa$. Let $n_l^{(h)}$ denote the number of subjects that are in cluster $h$ according to $g(\cdot)$ as well as in category $l$ given by $\kappa$. Normalized mutual information proposed by Strehl and Ghosh (2002) can be formulated as,

$$NMI = \frac{2}{n} \sum_{h=1}^{N} \sum_{l=1}^{K} n_l^{(h)} \log_{N,K} \left( \frac{n_l^{(h)} \cdot n}{n_l n^{(h)}} \right). \quad (3.18)$$

3.2.3 Criterion Function $J(x)$

Recall from Equation (3.2), our performance measure of each individual gene, $x$, is $J(x) = E\left\{ \phi \left( g(X^{(x)}), \kappa \right) \right\}$, the average performance measure of all subsets in $\mathcal{X}_d^{(x)}$. Let $|\mathcal{X}_d^{(x)}|$ be the number of subsets in $\mathcal{X}_d^{(x)}$. Then

$$|\mathcal{X}_d^{(x)}| = \binom{p - 1}{d - 1} = \frac{(p - 1)!}{(p - d - 2)!(d - 1)!},$$
which can be very large even for moderate values of $p$ and $d$. In practice it is impossible to enumerate all the subsets in $\mathcal{X}_d^{(x)}$. Instead we may bootstrap or sample from the space $\mathcal{X}_d^{(x)}$ and then estimate $J(x)$ by

$$
\widehat{J}_n(x) = \frac{1}{B} \sum_{i=1}^{B} \phi_n (g(X_i), \kappa), \quad X_i \in \mathcal{X}_d^{(x)},
$$

(3.19)

where $B$ is the bootstrap size.

There are several concerns about using $J_n(x)$ in (3.19) as an estimate of $J(x)$:

1). It samples from $\mathcal{X}_d^{(x)}$ without a regard of random errors. Also, although the external class label $\kappa$ is usually assessed by an expert, it may contain errors and mistakes, or simply have random variation.

2). Different sample class groups may have different number of samples.

3). Using all the samples in $X_i$, we would ‘overfit’ the data in hand.

Therefore we propose to improve $\widehat{J}_n(x)$ by replacing $\phi_n$ in (3.19) with

$$
\frac{1}{S} \sum_{j=1}^{S} \phi_n (g(X_{i,j}), \kappa),
$$

(3.20)

where each $X_{i,j}$ consists of some columns of $X_i$, which are subsampled randomly with a stratified sampling scheme. In the stratified sampling, the same number of samples (or columns) are drawn from each of column (or class) groups. This $X_{i,j}$ is called a subsample of $X_i$. The $X_{i,1}, X_{i,2}, \ldots, X_{i,S}$ are iid realizations of the subsample. Therefore, our final $J_n(x)$ is

$$
J_n(x) = \frac{1}{B} \sum_{i=1}^{B} \frac{1}{S} \sum_{j=1}^{S} \phi_n (g(X_{i,j}), \kappa), \quad X_i \in \mathcal{X}_d^{(x)},
$$

(3.21)
3.2.4 Feature Selection

The next step is to select our best set of genes based on the criterion function $J_n(x)$ defined in (3.21) and evaluated for each gene in $x_1, x_2, \ldots, x_p$. In principle we need to choose a cutoff point $c_J$ so that the selected genes satisfy the condition,

$$\{x_i : J_n(x_i) > c_J, i = 1, 2, \ldots, p\}.$$ 

Graphically $(x_i, J_n(x_i))$ can be plotted as a point in a 2-dimensional space. See Figure 3.1. There are usually a small proportion of genes clearly standing up from the rest. In this case $J_n(x)$ for the majority of non-interesting genes forms a central band and the small portion of interesting ones stand out from this majority as ‘outliers’. In order to separate the small proportion of differential genes from the majority of non-differential ones, we propose to use a fitness score, which is a robust Mahalanobis distance calculated using $J_n(x_i)$,

$$FIT(x_i) = \sqrt{(J_n(x_i) - T)'V^{-1}(J_n(x_i) - T)}, \quad i = 1, 2, \ldots, p,$$ 

(3.22)
Figure 3.2: Illustration of patterns for 2-dimensional $J_n(x)$

where $T$ and $V$ are the minimum covariance determinant (MCD) (Rousseeuw and Vanzomeren, 1990a,b) estimators of location and scatter of $J_n(x_i), i = 1, 2, \ldots, p$, respectively. MCD is a highly robust estimator of multivariate location and scatter, which can be efficiently computed with the FAST-MCD algorithm of Rousseeuw and Van Driessen (1999). We consider the $\chi^2$ distribution for the majority of $FIT^2$, which denotes the squared values of $FIT$ in (3.22). More considerations on the distribution of the robust Mahalanobis distance can be found in the work by Hardin and Rocke (2005). Since $FIT$ is essentially a distance measure for multi-dimensional data to its center, it can be used to group points defined by multiple class labels, say, $\kappa_1, \kappa_2, \ldots, \kappa_d$. For example, $\kappa_1$, may represent breast cancer status, and $\kappa_2$ may represent if a sample is from a coffee drinker or not. Thus, each resulting $J$ from $\kappa_i$ via (3.22) can be considered as a coordinate of d-dimensional space. For example, for $d = 2$, the majority $J$ will form a circle in the middle, and the interesting ones will stand outside of this circular center. This is shown in Figure 3.2.
3.2.5 RPselection Algorithm

Let \( X \) denote a \( p \times n \) data matrix measured on \( p \) genes from \( n \) samples. Our RPselection algorithm computes all \( J(x_i) \)'s once for all at its last step:

1) Select \( X_1 \) of size \( d \) randomly from \( X \) that contains all the \( p \) genes. In other words, \( d \) rows of \( X = (x_{ij})_{p \times n} \) are chosen to be \( X_1 \).

2) Obtain \( X_{1,1}, X_{1,2}, \ldots, X_{1,S} \) by independently subsampling the columns of \( X_1 \) such that number of columns from all groups (specified by \( \kappa \)) are equal. Then compute \( g(X_{1,1}), g(X_{1,2}), \ldots, g(X_{1,S}) \) by applying the pam.

3) Evaluate \( \phi_{n}(g(X_{1,1}), \kappa), \phi_{n}(g(X_{1,2}), \kappa), \ldots, \phi_{n}(g(X_{1,S}), \kappa) \) by (3.12):

\[
\phi_{n}(g(X), \kappa) \propto \min_{I} \left( \text{tr}(CII) \right),
\]

where \( C = 1 - U'W \) and \( U \) is determined by \( g \) and \( W \) by \( \kappa \).

4) Calculate \( \phi_{n}(g(X_1), \kappa) \) by (3.20):

\[
\phi_{n}(g(X_1), \kappa) = \frac{1}{S} \sum_{j}^{S} \phi_{n}(g(X_{1,j}), \kappa).
\]

5) Repeat steps 1) - 4) \((B - 1)\) times to obtain \( X_2, X_3, \ldots, X_B \), and \( \phi_{n}(g(X_2), \kappa), \phi_{n}(g(X_3), \kappa), \ldots, \phi_{n}(g(X_B), \kappa) \).

6) Evaluate the criterion function \( J \) for all genes \( x_i, i = 1, 2, \ldots, p \),

\[
J_{n}(x_i) = \frac{\sum_{j=1}^{B} \phi_{n}(g(X_{d,i}), \kappa) I(x_i \in X_d)}{\sum_{i=1}^{B} I(x_i \in X_d)}.
\]

7) Calculate \( FIT(x_i), i = 1, 2, \ldots, p \) using (3.22):

\[
FIT(x_i) = \sqrt{(J_{n}(x_i) - T)'V^{-1}(J_{n}(x_i) - T)}, \quad i = 1, 2, \ldots, p,
\]

8) Compute the p-value of \( FIT^2(x_i) \) based on a \( \chi^2_1 \) distribution to evaluate how extreme each gene \( x_i \) is from the cloud of the majority of genes.

9) Select the genes with the first \( k \) smallest p-values.
3.2.6 Choice of $d$, $B$ and $S$

One natural question we need to answer before we apply the RPselection algorithm is how to choose the parameters $d$ and $B$.

**Equal importance of features**

The microarray expression levels after a log transformation are approximately proportional to corresponding transcript abundance. Under the principle of equal importance of genes, the performance measure $J_n$ should be evaluated in a spirit of equal contribution on transcript abundance for all the genes. There is a good compliance to this principle if all genes have gone through the selection process described in the *RPselection algorithm* the same number of times. Let $n_x$ denote the number of times that the gene $x$ goes through the selection process. A small variance relative to the mean, i.e. a small CV of $n_x$, indicates that the number of times all genes have gone through the *RPselection algorithm* is about the same. It can be easily shown that $n_x \sim \text{Bin}(B, d/p)$. So, the CV is $\sqrt{1 - d/p} \sqrt{Bd/p} = \sqrt{(p/d - 1)/B} \rightarrow 0$, as $B \rightarrow \infty$ or $d \rightarrow p$. Simulation results shown in Figure 3.3 confirms that $CV \rightarrow 0$ as $d$ increases to $p$. Hence, all genes will be selected about an equal number of times when $d$ of the partition clustering procedure $g(\cdot)$ is relatively large. In Figure 3.3, $B = 10,000$, $p = 1000$, and $d$ varies from 1 to 1000.
Figure 3.3: CV of $n_x(d)$ vs $d$: coefficient of variation (CV) of the average number of times that a gene goes through the RPselection algorithm versus $d$
Minimum requirement of \(d\)

Let \(A\) be the event that each gene is selected at least once by the \(RP\) selection algorithm.

**Lemma 3.2.1.** Given \(d, p\) and \(B\), using the \(RP\) selection algorithm we have

\[
\Pr(A|d, p, B) = 1 - \sum_{i=1}^{p-d} (-1)^{i-1} \binom{i}{p} \left( \frac{d}{p} \right)^B.
\]

(3.23)

**Proof:** Let \(\Omega\) be the set of \(p\) features, \(\Sigma\) be a \(\sigma\)-field of subsets of \(\Omega\), which represents all subsets. Further let \(\Pr\) be a probability measure on \(\Sigma\). Denote \(A_{i_1, i_2, \ldots, i_m}\) the event that genes \(i_1, i_2, \ldots, i_m, m = 1, 2, \ldots, p\), are missing in all \(B\) subsamples of size \(d\) in (3.21), and denote the induced probability by \(B\) independent experiments to be \(\Pr^B\), then

\[
\Pr^B(A_{i_1}) = \left( \frac{p-1}{d} \right)^B, \quad i_1 = 1, \ldots, p,
\]

\[
\Pr^B(A_{i_1, i_2}) = \left( \frac{p-2}{d} \right)^B, \quad 1 \leq i_1 < i_2 \leq p,
\]

\[\vdots\]

\[
\Pr^B(A_{i_1, i_2, \ldots, i_{p-d}}) = \left( \frac{p-d}{d} \right)^B, \quad 1 \leq i_1 < \ldots < i_{p-d} \leq p,
\]
\[ Pr^B(A_{i_1, i_2, \ldots, i_{p-d+1}}) = 0, \quad 1 \leq i_1 < \ldots < i_{p-d+1} \leq p, \]
\[ Pr^B(A_{i_1, i_2, \ldots, i_{p-d+2}}) = 0, \quad 1 \leq i_1 < \ldots < i_{p-d+2} \leq p, \]
\[ \vdots \]
\[ Pr^B(A_{i_1, i_2, \ldots, i_p}) = 0, \quad 1 \leq i_1 < \ldots < i_p \leq p, \]

By the Inclusion-Exclusion relation, we have

\[ Pr(\overline{A}) = \sum_{l=1}^{p} (-1)^{l-1} \sum_{1 \leq i_1 < \ldots < i_l \leq p} Pr^B(A_{i_1, \ldots, i_l}). \tag{3.24} \]

Thus, the probability that each of the genes is selected at least once is,

\[ Pr(A) = 1 - Pr(\overline{A}) = 1 - \sum_{l=1}^{p} (-1)^{l-1} \sum_{1 \leq i_1 < \ldots < i_l \leq p} Pr^B(A_{i_1, \ldots, i_l}), \]

which is Equation (3.23). For \( p = 1000 \), we plot this probability as a function of \( d \) and \( B \) in Figure 3.4, where \( d \) varies from 1 to 200 and \( B \) varies from 1 to 2000. We can see from the plots that \( Pr(A | d, p, B) \) quickly reaches 1 as \( d \) and \( B \) increase. This is reasonable because more genes are included in the partition clustering procedure \( g(\cdot) \) and more bootstrap samples we use to evaluate the \( J_n \), more likely that every gene is selected at least once to go the evaluation process in the RPselection algorithm. Under this scenario, the best \( d \) and \( B \) are the ones that have \( Pr \) equal 1, which corresponds to the region with large \( d \) and large \( B \) in Figure 3.4.

Let \( d_0 \) be the smallest \( d \) s.t. \( Pr(A | d, p, B) = 1 \), then all \( d \) s.t. \( d \geq d_0 \) is called the working \( d \).
Figure 3.4: Probability that each gene selected at least once in the RPselection algorithm: The upper plot is the 3-D plot of Pr vs. d and B and the lower is the corresponding contour plot
Informative $d$ for evaluating $\phi$

Our objective is to find a $d$, s.t. the resulting $RP$selection algorithm can efficiently select the most informative genes. From this perspective we choose a $d$ value by looking at how variable the concordance measure $\phi$ is when different $d$’s are used. The concordance measure $\phi$ measures the concordance of the partition clustering procedure $g(\cdot)$ with respect to a known external class label $\kappa$. Clearly, the higher $\phi$ is the better the partition clustering procedure $g(\cdot)$ performs in partitioning the objects with respect to the external known class label $\kappa$. As shown in Figure 3.5, the more genes are included into the partition clustering procedure $g(\cdot)$, the higher $\phi$ is. Again as shown in Figure 3.5, starting from $d = 700$ or 800, more than half of the $\phi$ values are 1, which represents a perfect partitioning. However, the small variation of $\phi$ values (though large) are not informative in differentiating the genes in the $RP$selection algorithm. Therefore, good $d$ values are those s.t. both the median and range of $\phi$ are large. In Figure 3.5 a good choice of $d$ is
somewhere between 200 and 400.

**Choice of $B$ and $S$**

Generally, in terms of choice of $B$, the larger the better, which is shown later in both simulation studies and the real data example. However, $B$ is limited by computing power. So, we suggest choosing the largest workable $B$ values, say $B = 10,000$ for $p = 2000$. $S$ is the number of times we subsample the columns, it is also limited by computing power. In the simulated data and the real data example below, we use $S = 100$.

### 3.3 Simulation Studies

In order to validate our algorithm, an effective approach is to apply our method to data simulated with known data characteristics and known ground truth. However, creating simulated data without practical consideration is not helpful. For this reason, we incorporate the characteristics of real microarray experiments into our simulation experiments.

The source of microarray errors or their variability in microarray data comprises biological and experimental factors, such as biological and individual replication, sample preparation, hybridization and image processing. Moreover, one gene often shows quite heterogeneous error variability under different biological and experimental conditions. Therefore, the errors or their variability must be estimated separately across different factors or conditions.

In our simulation studies, we are interested in the statistical characteristics of the data that are related to the performance of our feature selection method. Without losing the generality, we will focus on simulating data with some differential expression patterns, which is also a widely adopted design in real microarray data experiments.
We will adopt the Bayesian hierarchical error model (HEM) proposed by Cho and Lee (2004) but incorporate some improvements by correcting a mistake pointed out by Wu et al. (2006).

### 3.3.1 Hierarchical Error Model (HEM)

Cho and Lee (2004) proposed a Hierarchical Error Model (HEM) to model many significant aspects of gene microarray data. One aspect is that error variability of gene expression data has two components, biological and experimental. The former reflects variation owing to biological sample heterogeneity, while the latter largely comes from the variation in sample preparation and sample design. The other important aspect is that the use of HEM is an effective way to represent complex relationships between parameters of interest imposed in modeling the microarray data.

Suppose that there are \( p \) genes on each microarray chip at each of two conditions, we observe the \( l \)-th gene expression value \( y_{i,j,k,l} \) of the \( i \)-th gene for a particular \( k \)-th individual with the \( j \)-th condition. Furthermore, assume the observations were log transformed. So,

\[
y_{i,j,k,l} = x_{i,j,k} + e_{i,j,k,l},
\]

\[
x_{i,j,k} = \mu + g_i + c_j + s_k + r_{i,j} + b_{i,j,k},
\] (3.25)

where \( e_{i,j,k,l} \) is the independent random error; \( \mu \) is the ground truth of the mean gene expression; \( g_i, c_j \) and \( s_k \) are the gene, condition and chip effects, respectively; \( r_{i,j} \) is the interaction effect of gene and condition; and \( b_{i,j,k} \) is the error term due to the biological variation. Here we assume,

\[
e_{i,j,k,l} \sim N(0, \sigma_{e,l}^2),
\]

\[
g_i \sim N(0, \sigma_{g_i}^2),
\]

\[
c_j \sim N(0, \sigma_{c_j}^2),
\]

\[
s_k \sim N(0, \sigma_{s_k}^2),
\]

\[
r_{i,j} \sim N(0, \sigma_{r_{i,j}}^2),
\]

\[
b_{i,j,k} \sim N(0, \sigma_{b_{i,j,k}}^2).
\] (3.26)

The biological variation, \( b_{i,j,k} \), varies on different expression levels, which is typical in practical microarray data. Therefore \( b_{i,j,k} \) is obtained by applying...
the local-pooled-error (LPE) baseline variance estimator presented in Jain et al. (2003) to some real microarray dataset. Based on LPE, biological variances corresponding to randomly chosen expression values are determined by a variance versus mean curve from a real microarray data.

In our simulation study, we choose,

$$(\sigma_e^2, \sigma_g^2, \sigma_c^2, \sigma_{rij}^2, \sigma_{bijk}^2) = (0.05, 0.01, 0.1, 0.5, 0.1, \text{LPE})$$

where LPE represents the baseline variance estimator from the local-pooled-error model, as discussed in Nykter et al. (2006); Cho and Lee (2004); Wu et al. (2006); Jain et al. (2003).

### 3.3.2 Simulated Data

Assume we have a total number of $p$ genes, and among them the number of differentially expressed ones is $k$. We set $p = 1000$ and $k = 25$. In order to simulate data using HEM, we need to define the grand mean $\mu$ in (3.25). We generate $\mu$ with a 2-group structure as shown in Figure 3.6.

The data contains two groups, corresponding to two conditions, so $j = 1, 2$ in (3.25). Let $\mu_1$ and $\mu_2$ denote the two vectors of 25 elements that represent the 25 differentially expressed genes in condition groups 1 and 2, respectively. Let $\mu_0$ denote the background mean of all the other $(p - 25)$ genes that are not differentially expressed.

$$\mu_1 = \begin{pmatrix} \mu_{1,1} \\ \mu_{1,2} \\ \vdots \\ \mu_{1,25} \end{pmatrix}, \quad \mu_2 = \begin{pmatrix} \mu_{2,1} \\ \mu_{2,2} \\ \vdots \\ \mu_{2,25} \end{pmatrix},$$
With real microarray data in mind we set,
\[
\mu_{1,i} = 11 - \frac{(i - 1) \times (11 - 9)}{24}, \quad i = 1, \ldots, 25
\]
\[
\mu_{2,i} = 8.3 + \frac{(i - 1) \times (12 - 8.3)}{24}, \quad i = 1, \ldots, 25
\]
\[
\mu_0 = 5, \quad \text{for all } i = 26, \ldots, 1000
\]
and then generate simulated data based on (3.26).

As an exploratory plot of expression intensities, many analysts prefer the $M$ vs. $A$ plot (MVA). For a pair of duplicate arrays, say $x_{i1}$ and $x_{i2}$, the MVA plot of $M = \log_2(x_{i1}/x_{i2})$ versus $A = \log_2 \sqrt{x_{i1}x_{i2}}$, $i = 1, \ldots, 1000$, can facilitate the investigation of between-array variability in terms of overall intensity. The MVA plot provides a very rough look of the data. As shown in Figure 3.7 for the simulated data, the biological variation varies on different expression levels, which closely assimilates variance patterns in practical microarray data. Meanwhile, as shown in the dendrogram in Figure 3.8, the simulated data retain the high concordances between samples. Here the
dendrogram is generated from a hierarchical clustering using a dissimilarity metric based on the Pearson correlation coefficient (3.3).

3.3.3 Performance Measures

In order to evaluate the performance of our RPselection algorithm, we will examine two properties: the ‘power’ of the algorithm in selecting the truly differentially expressed genes, and the accuracy of the algorithm in representing the order of the differentiation among those truly differentially expressed genes.

Property 1) Power in selecting the truly differentially expressed genes

The ‘power’ or the ‘true positive rate’, denoted as $q_{FIT}$, is estimated by the proportion of the truly differentially expressed genes that are correctly selected by the algorithm. Let $\mathcal{D}$ and $\overline{\mathcal{D}}$ denote the sets of indices for the truly differentially and non-differentially expressed genes, respectively. Then
a reasonable measure of the ‘power’ or ‘true positive rate’ is

$$q_{FIT} = \frac{\sum_{i=1}^{k} I(i(FIT) \in \mathcal{D})}{k},$$

(3.27)

where $k = |\mathcal{D}|$, the number of differentially expressed genes, $I(\cdot)$ is an indicator function, and $i(FIT)$ is the index of the gene that has the $i$th largest $FIT$ values,

$$f(x_{1(f)}) > f(x_{2(f)}) > \ldots > f(x_{p(f)}),$$

(3.28)

with $f = FIT$ and $x_{i(f)}$ denotes the gene that has the $i$th largest $FIT$ values. Generally $f$ in (3.28) is certain criterion function measuring the performance of the features. Another example of $f$ is the $t$ statistics in the $t$-test based algorithm. The value $q_{FIT}$ defined in (3.27) is bounded between 0 and 1, with 0 indicating the poorest performance, i.e. none of the differential genes are selected in the top-$k$ list and with 1 indicating the best possible performance that all $k$ differential genes compose the top-$k$ list.

**Figure 3.8: Dendrogram of the simulated Data**
Property 2) **Accuracy in representing the order of the differentiation**

To study how well the rank of the \(FIT\) values represents the magnitude of the gene differentiation, we can calculate the concordance of the ranks in \(FIT\) values with respect to the mean differences between two sample groups. A reasonable concordance measure based on Spearman’s rank correlation coefficient is

\[
R = 1 - \frac{6 \sum_{l=1}^{k} (i_l(FIT) - i_l(\Delta \mu))^2}{k(k^2 - 1)},
\]  

(3.29)

where \(i_1, i_2, \ldots, i_k\) are the indices of the truly differentially expressed genes, so \(i_1, i_2, \ldots, i_k \in \mathcal{D}\). Additionally \(i_l(FIT) \in \{1(FIT), 2(FIT), \ldots, p(FIT)\}\) with \(l = 1, 2, \ldots, k\), are defined as in (3.28) with \(f\) being the fitness measure \(FIT\). Similarly, \(i_l(\Delta \mu) \in \{1(\Delta \mu), 2(\Delta \mu), \ldots, p(\Delta \mu)\}\) with \(l = 1, 2, \ldots, k\) are defined with \(f\) being the difference in the means between two sample groups, so

\[
f(x_i) = \Delta \mu(x_i) = \mu_{2,i} - \mu_{1,i}, \quad i = 1, 2, \ldots, p.
\]  

(3.30)

### 3.3.4 Simulation Results

Figure 3.9 and Table 3.1 below show the median of the 25 estimates of \(q\), which are calculated by repeatedly applying the **RPselection algorithm** (Section 3.2.5) to randomly generated microarray data. As \(B\) and \(d\) increases, \(q\) increases. Figure 3.10 and Table 3.2 below show the IQR of 25 \(q\)’s. As \(B\) and \(d\) increases, \(IQR\) converges to zero. Therefore, our method eventually selects all the differentially expressed genes as the number \(B\) of the bootstrap samples increases to a large enough number. Meanwhile we can also see that the performance is different for different \(d\) values. Even through with any working \(d\) sooner or later, the algorithm will converge to the truth, with some \(d\)’s the **RPselection algorithm** converges faster than others. This phenomenon is consistent to the result shown Figure 3.5 of Section 3.2.6.

Tables 3.3, 3.4 and 3.5 show \(q\) values of 25 simulated microarray data.
generated randomly using the HEM model as described in Section 3.3.1. Clearly we can see:

1). As the number of bootstrap samples, $B$, increases, more truly differentially expressed genes are correctly selected into the top 25 gene list by our RPselection.

2). The variation of $q$ or the stability of the algorithm improves as $B$ increases.

3). Performance of the algorithm varies according to different $d$’s, but the algorithm converges to the same truly differentially expressed 25 genes regardless of different choices of $d$’s as long as we choose a large enough $B$.

4). Our approach outperforms the standard $t$-test based feature selection approach when $B$ is large enough, say $B \geq 5000$. The corresponding average $q$ values are smaller than those by the $RPselection$ procedure.

5). Both our approach and the $t$-test based approach gain improvement over larger sample size, as we can see by comparing the resulting $q$’s for different sample size $n = 15, 30$ and 45, as shown in Tables 3.3, 3.4 and 3.5, respectively.
Figure 3.9: Power $q_{FIT}$ of the RPselection algorithm: the proportion of truly expressed genes that are correctly selected by the algorithm for randomly generated microarray data. This is a diagrammatically representation of the data in Table 3.1

<table>
<thead>
<tr>
<th>$d$</th>
<th>1k</th>
<th>2k</th>
<th>5k</th>
<th>10k</th>
<th>20k</th>
<th>50k</th>
<th>100k</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>NA</td>
<td>0.28</td>
<td>0.36</td>
<td>0.44</td>
<td>0.44</td>
<td>0.36</td>
<td>0.36</td>
</tr>
<tr>
<td>20</td>
<td>0.32</td>
<td>0.52</td>
<td>0.68</td>
<td>0.76</td>
<td>0.76</td>
<td>0.84</td>
<td>0.84</td>
</tr>
<tr>
<td>40</td>
<td>0.52</td>
<td>0.72</td>
<td>0.92</td>
<td>0.96</td>
<td>0.96</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>60</td>
<td>0.64</td>
<td>0.8</td>
<td>0.92</td>
<td>0.96</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td>0.68</td>
<td>0.8</td>
<td>0.88</td>
<td>0.96</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>200</td>
<td>0.72</td>
<td>0.8</td>
<td>0.88</td>
<td>0.92</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>300</td>
<td>0.64</td>
<td>0.76</td>
<td>0.88</td>
<td>0.96</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3.1: Power $q_{FIT}$ of the RPselection algorithm: the proportion of truly expressed genes that are correctly selected by the algorithm for randomly generated microarray data. The data is diagrammatically presented in Figure 3.9, which are the median $q$ from 25 repeated applications of the algorithm refsec:alg upon the specific simulated microarray data.
Figure 3.10: IQR of \( q_{FIT} \) from 25 repetitions: the simulated microarray data is same as one used in Table 3.1. This is a diagrammatic representation of the data in Table 3.2.

<table>
<thead>
<tr>
<th>( d )</th>
<th>1k</th>
<th>2k</th>
<th>5k</th>
<th>10k</th>
<th>20k</th>
<th>50k</th>
<th>100k</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>NA</td>
<td>0.05</td>
<td>0.08</td>
<td>0.08</td>
<td>0.12</td>
<td>0.04</td>
<td>0.04</td>
</tr>
<tr>
<td>20</td>
<td>0.08</td>
<td>0.12</td>
<td>0.08</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
<td>0</td>
</tr>
<tr>
<td>40</td>
<td>0.08</td>
<td>0.08</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
<td>0</td>
</tr>
<tr>
<td>60</td>
<td>0.08</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>0.08</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>200</td>
<td>0.08</td>
<td>0.04</td>
<td>0.08</td>
<td>0.04</td>
<td>0.04</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>300</td>
<td>0.12</td>
<td>0.08</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.2: IQR of \( q \) from 25 repetitions: the simulated microarray data is same as one used in Table 3.1. This data is diagrammatically presented in Figure 3.10.
Table 3.3: Average and standard deviation of $q_F IT$, for $n = 15$: the entries are averages (with standard deviations in parentheses) of $q$ from 25 randomly generated microarray data using the HEM model; Correspondingly for the t-test, the average $q$ is 0.80 and the standard deviation is 0.05

<table>
<thead>
<tr>
<th>$d$</th>
<th>5k</th>
<th>10k</th>
<th>20k</th>
<th>50k</th>
<th>100k</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.90(0.03)</td>
<td>0.97(0.02)</td>
<td>0.99(0.02)</td>
<td>1.00(0.01)</td>
<td>1.00(0)</td>
</tr>
<tr>
<td>100</td>
<td>0.89(0.05)</td>
<td>0.95(0.03)</td>
<td>0.99(0.02)</td>
<td>1.00(0.01)</td>
<td>1.00(0)</td>
</tr>
<tr>
<td>200</td>
<td>0.86(0.06)</td>
<td>0.94(0.04)</td>
<td>0.97(0.03)</td>
<td>0.99(0.01)</td>
<td>1.00(0)</td>
</tr>
<tr>
<td>300</td>
<td>0.85(0.05)</td>
<td>0.92(0.04)</td>
<td>0.96(0.03)</td>
<td>0.99(0.03)</td>
<td>0.99(0.01)</td>
</tr>
</tbody>
</table>

Table 3.4: Average and standard deviation of $q_F IT$, for $n = 15$: the entries are averages (with standard deviations in parentheses) of $q$ from 25 randomly generated microarray data using the HEM model; Correspondingly for the t-test, the average $q$ is 0.84 and the standard deviation is 0.04

<table>
<thead>
<tr>
<th>$d$</th>
<th>5k</th>
<th>10k</th>
<th>20k</th>
<th>50k</th>
<th>100k</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.92(0.05)</td>
<td>0.98(0.03)</td>
<td>1.00(0.01)</td>
<td>1.00(0.00)</td>
<td>1(0.00)</td>
</tr>
<tr>
<td>100</td>
<td>0.88(0.05)</td>
<td>0.94(0.03)</td>
<td>0.98(0.03)</td>
<td>1.00(0.01)</td>
<td>1(0.01)</td>
</tr>
<tr>
<td>200</td>
<td>0.89(0.04)</td>
<td>0.95(0.04)</td>
<td>0.98(0.02)</td>
<td>1.00(0.00)</td>
<td>1(0.00)</td>
</tr>
<tr>
<td>300</td>
<td>0.84(0.04)</td>
<td>0.92(0.04)</td>
<td>0.97(0.03)</td>
<td>0.99(0.02)</td>
<td>1(0.00)</td>
</tr>
</tbody>
</table>

Table 3.5: Average and standard deviation of $q_F IT$, for $n = 45$: the entries are averages (with standard deviations in parentheses) of $q$ from 25 randomly generated microarray data using the HEM model; Correspondingly for the t-test, the average $q$ is 0.85 and the standard deviation is 0.04

<table>
<thead>
<tr>
<th>$d$</th>
<th>5k</th>
<th>10k</th>
<th>20k</th>
<th>50k</th>
<th>100k</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.91(0.04)</td>
<td>0.96(0.03)</td>
<td>1.00(0.01)</td>
<td>1.00(0.00)</td>
<td>1.00(0.00)</td>
</tr>
<tr>
<td>100</td>
<td>0.88(0.04)</td>
<td>0.93(0.03)</td>
<td>0.98(0.03)</td>
<td>1.00(0.01)</td>
<td>1.00(0.01)</td>
</tr>
<tr>
<td>200</td>
<td>0.87(0.04)</td>
<td>0.94(0.04)</td>
<td>0.99(0.02)</td>
<td>1.00(0.00)</td>
<td>1.00(0.00)</td>
</tr>
<tr>
<td>300</td>
<td>0.85(0.05)</td>
<td>0.93(0.04)</td>
<td>0.99(0.02)</td>
<td>1.00(0.00)</td>
<td>1.00(0.00)</td>
</tr>
</tbody>
</table>
3.4 Real Data Application

We apply the RP algorithm to a real microarray data.

3.4.1 Uveal Melanoma Data

The objective of this study is to find a reasonably accurate prognostic molecular classifier for the uveal melanoma using microarray technology. Twenty-seven patients were included in the study, whose tumor specimens were collected from eyes undergoing enucleation. The demographics (age, sex), tumor features (location, largest basal diameter, height), histopathological features (percent epitheliod component, presence or absence of matrix patterns) and outcomes (follow up duration, survival status [alive, dead due to metastasis, dead due to other causes]) were recorded for each patient. The data are pre-
Figure 3.12: Correlation-based clustering using all 3514 genes on all the samples in the uveal melanoma data

presented in Figures 3.11, 3.12 and 3.13. The colors green, red and blue in the plots indicate the survival status of alive, dead due to metastasis and dead due to other causes, respectively. The tumor cDNA array used in this study comprised a subset of sequence-verified cDNA clones from Research Genetics, Inc., 40,000-clone set representing approximately 4000 genes involved in tumorigenesis (Frevel et al., 2003; Li et al., 2005a,b). The 4000 cDNAs were selected based on their implication in metastasis and cancer development in general from the literature and from the Affymetrix cancer G110 array. The complete list of spotted cDNAs on this tumor array can be downloaded from the site: http://geacf.cwru.edu/geacf/geaaspotteddescriptions.shtml.

Hierarchical cluster analysis and Multidimensional scaling (Hastie et al., 2001) were used to investigate the grouping structure in the gene expression
Figure 3.13: Correlation-based multidimensional scaling using all 3514 genes on all the samples in the uveal melanoma data
data of the 27 tumor samples. Correlation based dissimilarity as described in the equation (3.3) is used as a dissimilarity metric (Eisen et al., 1998) in the hierarchical cluster analysis and the Multidimensional scaling. Hierarchical cluster analysis with the average linkage method (Hastie et al., 2001) is used to generate the dendrogram. Multidimensional scaling is also applied to investigate data grouping structure.

Here the goal is to select the genes that are most relevant to the outcome survival status when grouping the patients. After deleting the genes with missing values, we have 3514 genes in the final dataset. As demonstrated in the hierarchical clustering and the multidimensional scaling in Figures 3.12 and 3.13, respectively, the gene profile constructed with those 3514 genes effectively differentiate the 27 samples with respect to their survival status. The next question is to find the subset of the best relevant genes with a much smaller size that could separate the 27 samples well. To this end we apply our algorithm as described in Section 3.2.5.

3.4.2 Results

The performance score, \( J_n(x) \) in (3.21), and fitness score, \( FIT(x) \) in (3.22), of the uveal melanoma data are calculated and shown in Figure 3.14 and Figure 3.15, respectively. In Figure 3.14 the performance scores, \( J \), of all 3514 genes are plotted on the \( y \) coordinate, with all the genes listed in an arbitrary order as represented by the index number on the \( x \) coordinate. The significant genes based on their fitness scores with FDR adjusted p-values less then 0.05 are highlighted in red. We only selected the genes in the upper part, because only the genes that are positively relevant to the survival status are of interest in this study. The q-q plot of fitness scores versus \( \chi^2 \) quantiles are plotted in Figure 3.15.

In Figure 3.16, the heatmap of those significant genes with probe id, unigene id, order in fitness score and the common description from GeneBank
Figure 3.14: Performance score by applying the RPselection algorithm to the uveal melanoma data

are presented.
Figure 3.15: Fitness score $FIT^2$ for the uveal melanoma data
Figure 3.16: Selected key genes for the uveal melanoma data
Figure 3.17: Demonstration of $J_n(x)$ of the uveal melanoma data over different sets of $B$ and $d$

3.5 Discussion

Performance and fitness scores for different $B$ and $d$ of the uveal melanoma data are computed with results shown in Figure 3.17 and Figure 3.18. These results are consistent with the simulation results discussed in §3.3.4. The variation of the performance score $q$ decreases as the number of the bootstrap samples, $B$, increases. The selected top genes are invariant with respect to different choices of $d$. As further demonstrated in Figure 3.19, we can clearly see that, the larger the $d$, or the more genes we included in the **RPselection algorithm**, the higher the average performance score we get.
Figure 3.18: Demonstration of $FIT(x)$ of the uveal melanoma data over different sets of $B$ and $d$
Next, in order to demonstrate the application of our RPselection algorithm to real microarray data, we conducted the following experiment. We treat as the truth, the selected top genes from the parameter setting of \( B = 500,000, d = 200 \) and a threshold of 0.05 for FDR adjusted P values (call this true setting). We then vary \( B \) and \( d \), with \( B \) taking a value of 2000, 5000, 10,000, 50,000, 100,000 and \( d \) being different values of 100, 300, 600, 800, 1000. For each pair of \( B \) and \( d \), select the same number of top genes as selected from the true setting, and calculate \( q \) and \( R \) according to (3.27) and (3.29), respectively. The results for \( q \) are shown in Figure 3.20 and Table 3.6, and the results for \( R \) are shown in Figure 3.21 and Table 3.7.

Clearly if the top gene list from the true setting of \( B = 500,000 \) and \( d = 200 \) are treated as the truly differentially expressed genes and the amount of differentiation is in the order represented by this true setting, then for any working \( d \), the final selected gene list will be consistent with this true list, as \( B \) gets larger. This is indicated by the fact that \( q \) and \( R \) both approach one as \( B \) increases. See Figures 3.20 and 3.21 and Tables 3.6 and 3.7. As discussed
before, we see different choices of $d$ give different algorithm performance in terms of the convergence speed.
Figure 3.20: Diagrammatic presentation of $q$ values as in (3.27), computed by presumably treating the selected genes from $B = 500k$ and $d = 200$ as truth

<table>
<thead>
<tr>
<th></th>
<th>2k</th>
<th>5k</th>
<th>10k</th>
<th>50k</th>
<th>100k</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.11</td>
<td>0.33</td>
<td>0.62</td>
<td>0.76</td>
<td>0.8</td>
</tr>
<tr>
<td>300</td>
<td>0.11</td>
<td>0.18</td>
<td>0.36</td>
<td>0.84</td>
<td>0.89</td>
</tr>
<tr>
<td>600</td>
<td>0.07</td>
<td>0.18</td>
<td>0.35</td>
<td>0.78</td>
<td>0.82</td>
</tr>
<tr>
<td>800</td>
<td>0.11</td>
<td>0.15</td>
<td>0.25</td>
<td>0.67</td>
<td>0.73</td>
</tr>
<tr>
<td>1000</td>
<td>0.11</td>
<td>0.15</td>
<td>0.25</td>
<td>0.6</td>
<td>0.69</td>
</tr>
</tbody>
</table>

Table 3.6: Values of $q$ as in (3.27) computed by presumably treating the selected genes from $B = 500k$ and $d = 200$ as truth. The data in this table is also diagrammatically presented in Figure 3.20
Figure 3.21: Diagrammatic presentation of the $R$ values as in (3.29), computed by presumably treating the selected genes from $B = 500k$ and $d = 200$ as true.

<table>
<thead>
<tr>
<th>$d$</th>
<th>$2k$</th>
<th>$5k$</th>
<th>$10k$</th>
<th>$50k$</th>
<th>$100k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.64</td>
<td>0.68</td>
<td>0.76</td>
<td>0.78</td>
<td>0.77</td>
</tr>
<tr>
<td>300</td>
<td>0.55</td>
<td>0.61</td>
<td>0.7</td>
<td>0.86</td>
<td>0.85</td>
</tr>
<tr>
<td>600</td>
<td>0.43</td>
<td>0.51</td>
<td>0.62</td>
<td>0.7</td>
<td>0.75</td>
</tr>
<tr>
<td>800</td>
<td>0.44</td>
<td>0.5</td>
<td>0.69</td>
<td>0.76</td>
<td>0.73</td>
</tr>
<tr>
<td>1000</td>
<td>0.45</td>
<td>0.5</td>
<td>0.64</td>
<td>0.66</td>
<td>0.75</td>
</tr>
</tbody>
</table>

Table 3.7: Values of $R$ as in (3.29) computed by presumably treating the order of the selected genes from $B = 500k$ and $d = 200$ as truth. The data in this table is also diagrammatically presented in Figure 3.21.
Chapter 4

Software Packages

4.1 Introduction

Two software packages, *pfclust* and *rpselect*, are developed to provide software solutions to the methods described in Chapters 2 and 3. Packages *pfclust* and *rpselect* are implemented using R (http://www.r-project.org) and packed into portable R packages with R object-oriented programming framework. The R package provides a rich environment for data analysis and for statistical and bioinformatic computing (http://www.bioconductor.org). The current version of *pfclust* and *rpselect* can easily handle the data size of the real experiments and can be easily extended or customized by the rich computing and visualization functions in R.

Package *pfclust* generates a main class, *pfclust*, and provides the corresponding class methods in Chapter 2. Package *rpselect* creates a main class called *rpselect* for the calculated correspondence measure and provides the methods for feature selection based on the correspondence measure in Chapter 3.

The sections below are organized according to the two main software packages. Within each section, details about each software package are shown first, a quick reference card, in a standard R function format, about the main
classes, their methods, and other relevant R functions, is provided. Lastly
demos are provided by applying pfclust and rpselect to two real datasets,
the biomarker LOH data (Weber and Xu, 2007) and the microarray uveal
melanoma data (Singh, Sisley and Xu, 2007).

Since we use the same name for a class and its initiating method, we use
the italic font for the class and the typewriter font for the method in the
following sections to avoid confusion between classes and their methods.

4.2 Package pfclust

4.2.1 Features and Implementation

Package pfclust implements the PfCluster method discussed in Chapter 2.
The process of the pfclust is schematically presented into a flow chart shown
in Figure 4.1. In pfclust, an object of pfclust class is created by the method
pfclust which accepts a data frame as its input data, whose rows and
columns are the measurements corresponding to genes and samples. An
object of the class pfclust contains:

• the input data.
• hclust object from R hclust.
• PfCluster dendrogram that extends the dendrogram in R with more
  attributes and visualization features.
• a cohGroup object that contains the information about the resulting
  coherent groups, such as group id, group position in the PfCluster
dendrogram, members contained in each group and the coherence index
  of each group.
• a representative object that contains the representative profile of each
  coherent group. This representative profile is calculated either by av-
eraging all the profiles in the group or by a smoothing curve of all the
profiles with lowess or spline smoother.
Figure 4.1: Flow chart of software package pfclust
The method pfclust accepts a variety of options for customization, which include:

- a distance type option, which allows users to choose either $d$ in (2.3) or $D_{ii'}$ in (2.15) as the distance metric.
- a linkage method option for users to select the linkage method for the hierarchical clustering.
- a cutoff threshold $\alpha$ for estimating $d^*$ in (2.22).
- a distribution specification as in (2.15). $\chi^2$ distribution is used by default but users can choose to use $F$ distribution as well.
- a user-provided variance-covariance matrix for estimating $\Sigma_s$ as described in Section 2.6.
- convenient options for scaling the data before applying pfclust.
- other options to specify the smoothing parameters for group representative profiles.

Currently the generic class methods implemented for pfclust include:

- **plot**: to plot the PfCluster dendrogram allowing a variety of options to customize the plot, such as, options for skipping the sample labels, changing the figure title and specifying all the colors. It also provides a method to highlight certain coherent groups with different styles.
- **summary**: to provide a list of information from the resulting coherent groups, such as group id, the number of members and the coherent indexes.
- **subsetting**: to directly access the individual coherent group. This creates a new class called cohclust which includes group id, coherence index, PfCluster dendrogram, group representative and the data for this individual coherent group. For a cohclust class, users can use its class methods to directly access the members included in the coherent group and the number of members included. Users can also plot the dendrogram of this specific coherent group or plot its profile patterns.
For users who do not have an estimate of the variance-covariance matrix of the data, we suggest a two-step approach: first apply `pfclust` to the data using $d$ in (2.22); next obtain an estimate of variance-covariance matrix based on a pooled variance-covariance estimate of the clusters represented in the tree from the first-step application of `pfclust`. The clusters here are obtained by cutting the first-step PfCluster dendrogram at a height $h$ that minimizes $M$ if there is a minimum, otherwise, at a height $h$ where $M$ starts to climb up steeply to large values. This is done diagnostically by looking at the MVH plot. The formula for $M$ and the concepts of this diagnostic process have also been discussed in Section 2.6.2.

4.2.2 Quick Reference Card

```
**pfclust**  PfCluster Hierarchical Clustering.

pfclust(dat, dist.type = 2, linkage.method = "comp",
         cohIndex.cutoff = 0.05, x.vcov = NULL)

# s3 method for class 'pfclust'
plot(x, leaflab = "none",
     dendrogram.title = "Dendrogram",
     edgePar.base = list(col = "gray30", lty = "solid", lwd = 0.5),
     edgePar.cohGrps.base = list(col = "blue4", lty = "solid", lwd = 1),
     edgePar.cohGrps.highlight = list(col = "cyan3", lty = "solid", lwd = 1),
     highlight.p.col = "black", highlight.t.col = "white",
     grpid = NULL)

# S3 method for class 'pfclust'
summary(x)

# S3 method for class 'pfclust'
\[\](x,i)
```

ARGUMENTS
dat  a data frame, i.e. $X$ in Section 2.3, with rows and columns corresponding to genes and samples.

dist.type  an integer, 1 or 2, to specify the PfCluster distance such that

dist.type = 1 specifies

$$D_{ii'}(\alpha, \tau) = \sqrt{b_{ii'}^T b_{ii'}}$$

and dist.type = 2 specifies

$$D_{ii'}(\alpha, \tau) = \sqrt{b_{ii'}^T (C\Sigma_s C')^{-1} b_{ii'}},$$

where $b_{ii'} = C(X_i - X_{i'})$, $X_i$ is the $i$th row in data $X$ and $C$ is the contrast matrix defined in (2.1). Details about PfCluster distance can be found in Section 2.3 of Chapter 2.

linkage.method  a character specifying linkage method for the hierarchical clustering. It should be one of “ward”, “single”, “complete”, “average”, “mcquitty”, “median” or “centroid”.

cohIndex.cutoff  a numerical value between 0 and 1 specifying $\alpha$ for choosing $d^*$ of coherent clusters as in (2.22), which is recalled here,

$$\sum_{i<i'} P(D_{ii'}^2 > d) = \alpha.$$ 

where $d^*$ is the solution to the above equation for a given $\alpha$. We suggest $\alpha = 0.05$.

x.vcov  variance-covariance matrix. The default is $\text{var}(X)$.

leaflab  boolean with values TRUE or FALSE. It specifies whether or not to print sample labels.

dgePar*  a named list with values to specify edgePar for each component in the PfCluster dendrogram. edgePar.base specifies the base edgePar for the PfCluster dendrogram, edgePar.cohGrps.base specifies the base edgePar for the dendrograms of all the coherent clusters and edgePar.cohGrps.highlight specifies the edgePar for the dendrograms to be highlighted. See dendrogram in R for details on edgepar.
highlight*  a par option for plotting the coherent cluster(s) to be highlighted.

grpid    group id(s) to be highlighted.

METHODS

summary  printing and returning a summary of pfclust, which contains group ids, the number of members and coherent indices. Subsetting ‘[’ returns an object of class cohclust, which is the class representing a coherent cluster.

VALUE

a list containing the input data, a hclust object from R hclust, PfCluster dendrogram which extends the dendrogram in R, a list called cohGroups which contains the information about the resulting coherent clusters and a list called representative which contains the representative profile of each coherent cluster in the list cohGroups.

SEE ALSO

cohclust, hclust, dendrogram

EXAMPLES

pfc <- pfclust(x) # data.frame
plot(pfc) # plot the PfCluster dendrogram
summary(pfc) # give a summary of pfclust
pfc1 <- pfc[1] # create an object of class cohclust using subsetting

cohclust  Class of a coherent cluster from PfCluster

A class created by subsetting pfclust and representing a coherent cluster from PfCluster.

## S3 method for class 'cohclust'
plot(x,
    leaflab = 'none',
    dendrogram.title = 'Dendrogram',
    edgePar.cohGrps.base = list(col = 'blue4', lty = 'solid', lwd = 1),
    ...
)

## S3 method for class 'cohclust'
plotprofile(x,
    plotRepresentatives = T,
    plotIndivProfile = T,
    rep.line.col='cyan3',
    rep.line.lty=1,
    rep.line.lwd=2,
    ...
  )

## S3 method for class 'cohclust'
members(x)
## s3 method for class 'cohclust'
nmembers(x)
## method for class 'pfclust'
summary(x)

ARGUMENTS

  x  an object of class cohclust.

  leaflab  a character string to specify whether to print labels for the members. See leaflab in dendrogram in R for details.

  dendrogram.title  the title of the plot

  edgePar.cohGrps.base  the edgePar for the dendrogram of cohclust.

  ...  other arguments passed to plot.dendrogram

  plotRepresentatives  a boolean value of TRUE or FALSE asking whether to plot the representative profile.

  plotIndivProfile  a boolean value of TRUE or FALSE regarding whether or not to plot all individual data profiles. If FALSE it will only plot the representative profile. plotRepresentatives and plotIndivProfile can not both be FALSE.

  rep*  a par option for plotting representatives.

METHODS

  plot  plotting the branch of the dendrogram that corresponds to a coherent cluster.

  plotprofile  plotting all the profiles in the coherent cluster. It also gives
an option of plotting the representative profile of the coherent cluster.

**summary** printing and returning a summary of the coherent cluster with columns *group ids*, *number of members* and *coherent indexes*.

**members** listing all the members contained in this coherent cluster.

**nmembers** outputting the total number of members in the coherent cluster.

**VALUE**

class ‘cohclust’ containing the input data, the PfCluster dendrogram, and representative of the corresponding specified coherent cluster. The class is also attached with two attributes, *group id* and *coherence index*.

**SEE ALSO**

pfclust, hclust, dendrogram,

**EXAMPLES**

```r
pfc <- pfclust(x) # x is a data.frame
pfc1 <- pfc[1] # crates a cohclust by subsetting pfclust
plot(pfc1)
plotprofile(pfc1)
members(pfc1)
```

**pfcluster.M** Compute M’s of a dendrogram

```r
pfcluster.M(d, dat, h = NULL, n.node = 25)
## S3 method for class ‘Mvh’
plot(M)
```

**ARGUMENTS**

- **d** a dendrogram.
- **dat** a data matrix used for constructing the dendrogram *d*.
- **h** the lower bound of height *h* to cut the tree. The resulting clusters are used to calculate the pooled variance-covariance estimate.
- **n.node** the number of top nodes to be used for calculating the pooled variance-covariance estimate. It will not be used if *h* is specified. The default is to use the top 25 nodes in the dendrogram *d*.
METHODS
plot plotting $M$ vs $h$.

VALUE
A class $Mvh$ with components $d.heights$, $d.Ms$ and $d.inf$. $d.heights$ contains the heights at which the dendrogram is cut. $d.Ms$ are the corresponding $M$’s obtained at the corresponding $d.heights$, calculated by,

$$M = \sum_{i=1}^{p} \left( \nu \log s_i^2 - \sum_{j=1}^{q} \nu_j \log(s_{ij}^2) \right)^2$$

d.$inf$ contains the minimum $M$ and the corresponding $h$. Other details of each term are discussed in Section 2.6.2 of Chapter 2.

SEE ALSO
 pfcluster.estVar

EXAMPLES

```r
pfc <- pfclust(x) # x is a data.frame
M <- pfcluster.M(pfc$dendrogram)
plot(M)
```

| pfcluster.estVar | Compute $M$ by cutting the dendrogram at a specific height |

pfcluster.estVar(d, dat, h)

ARGUMENTS
- `d` a dendrogram.
- `dat` a data matrix used for constructing the dendrogram `d`.
- `h` height to cut the tree. The resulting clusters are used to calculate the pooled variance-covariance estimate of data.

VALUE
- a $M$ value.
SEE ALSO
pfcluster.M

EXAMPLES

```r
pfc <- pfclust(x) # x is a data.frame
M <- pfcluster.estVar(pfc$dendrogram,h=1)
```

4.2.3 Demonstration

In this section we show how to apply the package `pfclust` to the biomarker LOH data (Weber and Xu, 2007).

Load the `pfclust` package and the LOH data

```r
> library(pfclust)
> data(loh)
> str(loh)
List of 2
$ S: 'data.frame': 364 obs. of 4 variables:
 ..$ stage1: num [1:364] 0.438 0.625 0.375 0.250 0.188 ...
 ..$ stage2: num [1:364] 0.304 0.217 0.478 0.304 0.304 ...
 ..$ stage3: num [1:364] 0.324 0.441 0.353 0.294 0.324 ...
 ..$ stage4: num [1:364] 0.289 0.474 0.421 0.289 0.447 ...

$ T: 'data.frame': 364 obs. of 4 variables:
 ..$ stage1: num [1:364] 0.375 0.500 0.375 0.125 0.312 ...
 ..$ stage2: num [1:364] 0.348 0.609 0.348 0.304 0.391 ...
 ..$ stage3: num [1:364] 0.382 0.471 0.500 0.324 0.324 ...
 ..$ stage4: num [1:364] 0.316 0.579 0.368 0.237 0.500 ...
```

Apply the MIP filtering described in Section 2.8.1.

For this data, it is not necessary to use any filtering to reduce the data, but we apply it anyway to demonstrate how filtering works in the `pfclust` package.

```r
> pct2keep <- .5
> loh.s <- pfc.filter(loh$S, pct2keep, max, median)
> heatmap(as.matrix(loh.s),
```
The resulting heatmap is shown in Figure 4.2. In the following section, we apply Pfclust to the LOH data in a two-step process. The first step is to determine an initial estimate of the data variance and the second step is for an enhanced PfClust analysis which uses the variance estimated in the first step.

**First step: estimate data variance-covariance matrix**

1). Apply pfclust to the LOH data.

```r
> pfc.loh <- pfclust(loh.s,
+ Rowv=NA,
+ Colv=NA,
+ col=heatColors(51),
+ labRow="")
```
Figure 4.3: First step application of PfCluster to the filtered biomarker LOH data

-->Note: Root is coherent

> plot(pfc.loh, leaflab="none")

2). Diagnose using $Mvh$ plot. Figure 4.3 is produced using the plot method above and it shows the resulting tree structure of the 1st step application of PfCluster approach. Next we move onto estimating the data variance based on this 1st-step tree structure. The details have been illustrated before in Section 2.6.

> $Mvh \leftarrow$ pfcluster.M(pfc.loh$dendrogram,loh.s)
> $Mvh$

$h$

[1] 0.6942758 0.6132364 0.5911903 0.5107485 0.4733755 0.4072597 0.3984068 0.3902197 0.3712811 0.3700811 0.3384987 0.3332212 0.3264556 0.3184899 0.3027588 0.2678044 0.2533825 0.2530810 0.2477384 0.2442599 0.2305028 0.2271650 0.2170606 0.2131402 0.2046716

$M$

Figure 4.4: M-vs-h plot from the 1st step application of PfCluster to the filtered biomarker LOH

\[
\begin{array}{cccccccc}
\text{Inf} & \text{Inf} & \text{Inf} & \text{Inf} & \text{Inf} & \text{Inf} & \text{Inf} & \text{Inf}
\end{array}
\]

$S.diag$

\[
[1] 0.014071326 0.013350022 0.013080127 0.012645259 0.011838721
0.011010311 0.010349887 0.009756796 0.009461737 0.009151856
[12] 0.009066359 0.008838192 0.008885585 0.008854658 0.008806053
0.008793534 0.008770457 0.008643595 0.008551185 0.008461255 0.008368881
[23] 0.008430757 0.008434198 0.008312373
\]

$d.inf$

\[
\begin{array}{c}
d.inf \\
0.6942758 15.1920366
\end{array}
\]

attr("class")

[1] "Mvh"

> plot(Mvh)

Figure 4.4 shows no optimal $h$ associated with a minimum $M$. We choose the cutoff $h = 0.5$ because it appears in the $Mvh$ plot that $M$
Figure 4.5: The improved 2nd step application of PfCluster to the filtered biomarker LOH

starts to climb up to large values after 0.5. As described in Section 2.6.2, we cut the tree from the 1st step application of PfCluster algorithm and use a pooled variance estimate on the resulting clusters as the final variance estimate for the next step PfCluster analysis, which is shown below.

3). Estimate data variance-covariance

> var.M <- pfcluster.estVar(pfc.loh$dendrogram,loh.s,h=0.5)

2nd-step: apply pfclust with the estimated variance-covariance

> pfc.loh.update<- pfclust(loh.s,  
+     dist.type=2,  
+     cohIndex.cutoff=0.05,  
+     linkage.method="comp",  
+     x.vcov=var.M)  
> plot(pfc.loh.update,leaflab="none")
Other convenient class methods

```r
> loh.1 <- pfc.loh.update[1]
> plot(loh.1)
> plotprofile(loh.1)
```

Other convenient methods are provided by the pfclust package for looking at each of the resulting coherent clusters. In Figures 4.6, we show the dendrogram and profiles of the coherent cluster 1.

```r
> summ.loh <- summary(pfc.loh.update)
```

Number of coherent clusters: 19

Summary:

<table>
<thead>
<tr>
<th>n_members</th>
<th>CohIndex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.000000000</td>
</tr>
<tr>
<td>2</td>
<td>0.14846053</td>
</tr>
<tr>
<td>3</td>
<td>1.000000000</td>
</tr>
<tr>
<td>4</td>
<td>0.33517208</td>
</tr>
<tr>
<td>5</td>
<td>1.000000000</td>
</tr>
<tr>
<td>6</td>
<td>0.52183772</td>
</tr>
<tr>
<td>7</td>
<td>1.000000000</td>
</tr>
<tr>
<td>8</td>
<td>0.96337001</td>
</tr>
<tr>
<td>9</td>
<td>0.21394468</td>
</tr>
<tr>
<td>10</td>
<td>0.07269548</td>
</tr>
<tr>
<td>11</td>
<td>1.000000000</td>
</tr>
<tr>
<td>12</td>
<td>0.12718491</td>
</tr>
<tr>
<td>13</td>
<td>0.54522683</td>
</tr>
<tr>
<td>14</td>
<td>0.17504891</td>
</tr>
<tr>
<td>15</td>
<td>0.20747981</td>
</tr>
<tr>
<td>16</td>
<td>1.000000000</td>
</tr>
<tr>
<td>17</td>
<td>0.14413396</td>
</tr>
<tr>
<td>18</td>
<td>0.79642422</td>
</tr>
<tr>
<td>19</td>
<td>0.41090434</td>
</tr>
</tbody>
</table>
```

We can see from the above summary of the final clustering that there are a total of 18 coherent clusters with their coherence indexes listed. The profiles
Figure 4.6: The coherent cluster 1 in the improved PfCluster dendrogram (top) and its profiles (bottom)
of the first 4 coherent clusters are shown in Figure 4.7.

```r
> heatmap(as.matrix(loh.s),
+ Rowv=as.dendrogram(pfc.loh.update$hclust),
+ Colv=NA,
+ col=heatColors(51),
+ labRow="")
```

The heatmap in Figure 4.8 is generated with the code above. It is different from the one in Figure 4.2 in that the biomarkers in the former are ordered by the final dendrogram produced by `pfclust`, while in the latter biomarkers are not ordered.

### 4.3 Package `rpselect`

#### 4.3.1 Features and Implementation

Package `rpselect` implements the `RPselection algorithm` developed in Chapter 3. The steps of `rpselect` are shown in a flow chart presented in Figure 4.9. The
method \texttt{rpselect} generates the class \texttt{rpselect}. This method allows users to input a \texttt{data.frame} as data input, which is required to have rows and columns correspond to genes and samples, respectively. This input data has the same structure as that in \texttt{pfclust}. Method \texttt{rpselect} then allows users to specify the parameters required by the \texttt{RPselection algorithm}, such as $d, B, k, \kappa$ and $S$, as described in Section 3.2.5.

Users also have the option to choose one of the correspondence measures, $\phi$, and one of the profile distance measures for the partition clustering process. The correspondence measures provided in \texttt{rpselect} include the default correspondence measure defined in Section 3.2.2, \textit{Rand Index}, and \textit{Mutual Information} discussed in Section 3.2.2.

The profile distances users can specify are the Pearson, Kendall or Spearman correlation coefficients, the standardized Euclidean distance, the PfCluster profile distance defined in (2.10) or just the Euclidean distance.

Users can also specify the parameters for a \textit{leave-k-out} subsampling process described in (3.20). This subsampling process can be applied by leaving
**Figure 4.9:** Flow chart of software package rpselect
out a proportion of all the samples in the data, an equal proportion of samples in each group defined by the external class label $\kappa$, or an exact number of samples in each group, each time.

Several other class methods are implemented for the class `rpselect`. `select.features` does the gene selection based on FIT values, `plot.J` and `plot.FIT` plot the corresponding $J$ and FIT values for visual inspection, respectively. Finally a `toplist` method is provided to print out the top genes produced by `select.features`, where a user-specified cutoff threshold can be taken.

### 4.3.2 Quick Reference Card

```
RPselect(dat, d = 300, B = 100, k, classlabel, sample.n = 100,
        method.phi = c("correspondence", "Rand", "cRand",
                      "NMI"),
        method.dist = c("pearson", "kendall", "spearman",
                        "standardizedEuclid", "pfcluster", "euclidean"),
        leave.k = 1)
```

**ARGUMENTS**

- **dat**  data.frame, denoted as $X$ for later usage, with rows and columns corresponding to genes and samples.
- **d, B** the feature subset size and bootstrap size, respectively. Those are required for calculating $J_n(x)$, which is

  $$J_n(x) = \frac{1}{B} \sum_{i=1}^{B} \phi_n (g(X_i), \kappa), \quad X_i \in X_d^{(x)},$$

  where $\phi_n (g(X_i), \kappa)$ is the correspondence measure discussed in Section 3.2.2.
- **k**  the number of classes defined by the external class label $\kappa$.  

classlabel  a factor vector specifying the external class label $\kappa$

sample.n  the number of samples for subsampling step. It is denoted as $S$ in

$$
\phi_n(g(X_i), \kappa) = \frac{1}{S} \sum_{j} \phi_n(g(X_{i,j}), \kappa)
$$

method.phi  a character string specifying $\phi_n(x)$, it should be one of ‘correspondence’, ‘Rand’, ‘cRand’ or ‘NMI’, see details in Section 3.2.2 of Chapter 3.

method.dist  a character string specifying the distance measure used in hierarchical clustering process of the RPselection algorithm. It should be one of “pearson”, “kendall”, “spearman”, “standardizedEuclid”, “pfcluster” or “euclidean”.

leave.k  the number of samples left out in the subsampling step of the RPselection algorithm.

VALUE
a class of rpselect containing phi.matrix, J and positives. phi.matrix contains all the $\phi_n(x)$ values evaluated by RPselection algorithm. $J$ is the $J_n(x)$ for every feature and positives is a variable indicating whether a specific $J_n(x)$ is above or below the median of all $J_n(x)$’s.

SEE ALSO
select.features, plot.J, plot.FIT, toplist

EXAMPLES

classlabel <- c(1,1,2,1,2,1,2,2,1,2,2,1)  
rp <- rpselect(x, d=200,B=2000,k=2,classlabel,sample.n=200,leave.k=1)

select.features(x,cutoff.p = 0.05)

ARGUMENTS

x  an object of class rpselect.
cutoff.p  a numerical value between 0 and 1 specifying the feature selection
criteria, by default choosing 0.05

VALUE
a list of two components: fulllist, which includes a full list of all features
with their $J$ and FIT values; selectedlist, which is the selected list from the
full list based on the user-provided selection criteria (0.05).

SEE ALSO
rpselect, plot.J, plot.FIT, toplist

EXAMPLES

```r
classlabel <- c(1,1,2,1,2,1,2,2,1,2,2,1,2)
rp <- rpselect(x, d=200,B=2000,k=2,classlabel,sample.n=200,leave.k=1)
key.feature <- select.features(rp)
```

plot.J Plot J values from the RPselection algorithm.

```
plot.J(x,fs)
```

ARGUMENTS

- x  an object of class `rpselect`
- fs  an object from `select.features`

VALUE
None

SEE ALSO
rpselect, select.features, plot.FIT, toplist

EXAMPLES

```r
classlabel <- c(1,1,2,1,2,1,2,2,1,2,2,1,2)
rp <- rpselect(x, d=200,B=2000,k=2,classlabel,sample.n=200,leave.k=1)
keyf <- select.features(rp)
plot.J(rp,keyf)
```
\texttt{plot.FIT} \hspace{1em} Plot FIT values from the RPselection algorithm.

\texttt{plot.FIT(fs)}

ARGUMENTS

\hspace{1em} \texttt{fs} \hspace{1em} an object from \texttt{select.features}

VALUE

\hspace{1em} None

SEE ALSO

\texttt{rpselect, select.features, plot.J, toplist}

EXAMPLES

\begin{verbatim}
classlabel <- c(1,1,2,1,2,1,2,2,1,2,2,1,2)
rp <- rpselect(x, d=200,B=2000,k=2,classlabel,sample.n=200,leave.k=1)
keyf <- select.features(rp)
plot.FIT(rp,keyf)
\end{verbatim}

\texttt{toplist} \hspace{1em} list the top features selected from the RPselection algorithm.

\texttt{toplist(fs)}

ARGUMENTS

\hspace{1em} \texttt{fs} \hspace{1em} an object from \texttt{select.features}

VALUE

\hspace{1em} a \texttt{data.frame} with \textit{gene id}, \textit{FIT}^2, \textit{pvals} and \textit{positive}.

SEE ALSO

\texttt{rpselect, select.features, plot.J, plot.FIT}

EXAMPLES

\begin{verbatim}
classlabel <- c(1,1,2,1,2,1,2,2,1,2,2,1,2)
rp <- rpselect(x, d=200,B=2000,k=2,classlabel,sample.n=200,leave.k=1)
keyf <- select.features(rp)
toplist(keyf)
\end{verbatim}
4.3.3 Demonstration

In this example we use the uveal melanoma data to demonstrate the RPselection algorithm.

load \texttt{rpselect} and the data.

\begin{verbatim}
> Library(rpselect)
> data(uveal)
> names(uveal)
[1] "dt.foldchange" "array.info" "array.class"
> attach(uveal)
\end{verbatim}

Apply RPselection algorithm

\begin{verbatim}
> dn <- 400
> Bn=100
>
> rp.demo <- rpselect( d=dn,
+ B=Bn,
+ k=2,
+ dat=dt.foldchange,
+ classlabel=array.class,
+ sample.n=100,
+ leave.k=1
+ )
>
\end{verbatim}

Here \textit{dn}, \textit{Bn} and \textit{sample.n} correspond to the \textit{d}, \textit{B} and \textit{S} defined in the RPselection algorithm in Section 3.2.5, which are the size of the feature subset, the bootstrap size for evaluating $J_n(x)$ as shown in (3.21) and the resample size \textit{S} for evaluating $\phi_n(x)$ in (3.20).

Feature selection and graphics

\begin{verbatim}
> key.f <- select.features(rp.demo,cutoff.p=0.05)
\end{verbatim}
This function calculates FIT and the corresponding p values, and selects the interesting features with p-values < cutoff.p.

> plot.J(rp.demo,key.f)
> plot.FIT(key.f)

These two plot methods illustrate \( J_n(x) \) and FIT obtained from the RPselection algorithm, which are shown in Figure 4.10 and Figure 4.11, respectively.

> toplist(key.f)

genes   FIT2   pvals positive
1467 20194 46.588744 8.756179e-12   TRUE
1538 20279 24.992377 5.755744e-07   TRUE
1333 20039 24.280885 8.326126e-07   TRUE
2016 20820 19.271363 1.133944e-05   TRUE
2611 21503 18.745786 1.493530e-05   TRUE
1410 20128 17.588199 2.742851e-05   TRUE

**Figure 4.10:** Demonstration of plot method for \( J_n(x) \) in (3.21)
Figure 4.11: Demonstration of plot method for FIT(x) in (3.20)
Method `toplist` gives a list of genes with top p-values. Column `genes` is gene ID in the original dataset, `FIT2` is the squared `FIT` value, `pvals` is the p-value based on a $\chi^2$ distribution of $FIT^2$, and `positive` is an indicator of whether the interesting features are on the upper part or lower part in Figure 4.10.
4.4 Future Work

For pfclust and rpselect to work efficiently with real, large-volume genomic data, further optimization of the algorithm and the program structure are required. The RPselection algorithm is computationally intensive. A version based on the parallel computing framework has been built, but needs a stable mechanism to be packaged into a portable format. Currently it is only available on a customized parallel computing infrastructure. Additionally, the diagnostic tools discussed for choosing d’s in the RPselection algorithm need to be included. More convenient methods are expected to provide a better user-friendly interface. Finally the code needs to be improved for better structure and easy references.
Chapter 5

Discussion and Future Work

5.1 Discussion and Connection

We have entered a genomics and genetics era in which the generation of information outpaces our ability to understand its implication for prevention, diagnosis, and treatment of common multifactorial human diseases.\(^1\) The availability of vast amount of genetic information has presented us with important computational and statistical challenges.

One challenge is the variable selection problem. Standard variable selection procedures may not be applicable when \(p\), the number of variables, is large. The test-based gene-by-gene (or variable-by-variable) variable selection approach clearly does not take into account of the interactions among variables when selecting a set of important genes. There has been growing realization that interactions among multiple genetic and environmental factors are more important than any one factor for predicting the risk of common multifactorial diseases. Therefore, in disease etiology, we need to consider combinations of genetic variations or gene expression variables rather than

\(^1\)A multifactorial disease is a disease that is influenced by multiple environmental and genetic factors
one variable at a time in our genetic analysis. In studies involving diseased human tissues there are often predefined disease/biological classes, which should also be used together with the gene expression variables to find important genes. Identifying these important genes is helpful for developing effective diagnostic tools and treatments. Our RPselection algorithm is developed in consideration of interactions of genes and for both small and large values of \( p \). It selects a small set of genes that are most relevant to a given pre-defined disease, by resampling and partition clustering. It evaluates the relevance of each gene in a group with others, rather than in isolation. This is an important improvement over the current standard gene-by-gene based approach.

Another challenge is developing effective clustering procedures for gene profiles, so that these clusters are biologically “coherent”. In our PfCluster, the agglomerative hierarchical clustering is used as a data mining tool. A new class of distance metrics are proposed to measure biological “similarities” between data profiles. A coherence index is introduced first time to measure the internal quality of clusters. On both simulated and real data examples, PfCluster procedure worked remarkably well. It overperforms the standard correlation coefficient based clustering procedures. PfCluster opens a beginning for statistically evaluating clusters and for determining the number of clusters.

The RPselection algorithm and the PfCluster method can be used together in a data mining analysis. As a demonstration, the uveal melanoma data used in Chapter 3 is analyzed again here. The first step is to perform feature selection using the RPselection algorithm. Genes that are most relevant to ‘recurrence’ and ‘non-recurrence’ classes of the sample are selected (Figure 5.1). The second step is to apply the PfCluster method to group the selected genes based on their profiles, where we treat all 27 samples in the data as profile conditions. Based on the coherence index we find 10 coherent
groups (Figure 5.2). In Figure 5.3, we plot coherent group 1 and coherent group 2 of those 10 resulting groups. Coherent group 1 contains two profiles corresponding to two representative probes of a same unigene\(^2\). As shown in the upper part of Figure 5.3, probe ‘21453’ and probe ‘20194’ are highly ‘co-expressed’. In Figure 5.3, indices 1 - 11 represent subjects who are alive and 12-27 represent subjects who are dead due to metastasis, it is obvious from the profiles that the gene represented by probe ‘21453’ and ‘20194’ are highly downregulated in the subjects who are dead. In coherent group 2, probe ‘22277’ and ‘20672’ correspond to one unigene, and probe ‘20266’ represents a different gene and is co-expressed with the unigene represented by the probes ‘22277’ and ‘20672’. Probe ‘22277’, ‘20672’ and ‘20266’ are slightly upregulated in the dead subjects. As demonstrated by this example, the RPselection algorithm can efficiently select a small set of genes, the PfCluster method can be used to find a number of small, tight and important clusters.

\(^2\) An NCBI database entry. Each Unigene entry represents a set of transcript sequences that appear to come from the same transcription locus (gene or expressed pseudogene), together with information on protein similarities, gene expression, cDNA clone reagents, and genomic location.
Figure 5.1: Feature selection with RPselection
Figure 5.2: Profile cluster analysis and visualization with PfCluster
Figure 5.3: Coherent groups from PfCluster
5.2 Future work

Improved Approximation to $d^*$

As defined in Definition 2.4.3, a critical value $d^*$ at level $\alpha$ for a coherent cluster is the solution $d$ to

$$P(\max (D_{ii}^2) > d) = \alpha.$$  \hfill (5.1)

See (2.20). The values $d_1^*$ in (2.23) and $d_2^*$ by solving (2.38) are the first and second order approximations to $d^*$. For different values of $p$ and $\alpha$, $d_1^*$ and $d_2^*$ perform differently. As shown in Figures 2.4 and 2.5, $d_1^*$ is more biased in approximating the $d^*$ for larger $p$ or higher $\alpha$. $d_2^*$ is more accurate than $d_1^*$ when $p$ and $\alpha$ are small. $d_2^*$ is numerically unattainable when $p$ or $\alpha$ is too large.

Recall from (2.22) and (2.23) that $d_1^*$ is the solution to

$$N_0 P(D_{12}^2 > d) = \alpha,$$  \hfill (5.2)

and $d_2^*$ is the solution to equation (2.27), which is also given below,

$$N_0 P(D_{12}^2 > d) - N_1 (P(D_{12}^2 > d))^2$$

$$- N_2 P(D_{12}^2 > d, D_{13}^2 > d) = \alpha.$$  \hfill (5.4)

If we let $Pr$ denote $P(\max (D_{ii}^2) > d^*)$ for a given estimate of $d^*$, then we expect to see that $Pr$ be close to $\alpha$ if the estimate of $d^*$ is good enough. For the second order approximation as shown in (5.3), we have

$$Pr = S1 - S2a - S2b,$$

where,

$$S1 = N_0 P(D_{12}^2 > d), \quad \text{first order Pr}$$

$$S2a = N_1 (P(D_{12}^2 > d))^2,$$

$$S2b = N_2 P(D_{12}^2 > d, D_{13}^2 > d).$$
In Figure 5.4, black, red, green and blue lines represent $Pr, S1, S2a,$ and $S3b$ changing over $d$, respectively. As we can see from this figure, $S1$, the first order approximation, is a good representative of $Pr = S1 - S2a - S2b$, when $\alpha$ is very small. While terms $S2a$ and $S2b$ can improve the estimation of $Pr$ when $\alpha$ and $p$ are small, $Pr$ is obviously distorted when $\alpha$ is high or $p$ is large. This indicates that the higher order terms are required for an accurate approximation to the actual $P(\text{max}(D_{ii}'^2) > d)$ when $\alpha$ is high or when the number of profiles in a cluster, $p$, is large. Part of future work for PfCluster will be to find the higher order approximation to $d^*$. 

**Variance Estimation**

We shall also explore alternative approximation method to $d^*$. The method for deriving the critical value of Tukey’s all-pairwise simultaneous confidence intervals may be generalized to our multi-dimensional case to compute $d^*$. 

*Figure 5.4: Approximate $d^*$ using $d_2^*$*
In addition, $\alpha$ and $\tau$ in our derivation of the null distribution are considered fixed. If $\alpha$ needs to be estimated, the resulting $\tilde{D}_{ii}'^2$ is proportional to the sum of weighted squares of residuals in fitting an appropriate (auto) regression model. Thus, the limiting distribution to $\max_{ii'} D_{ii'}^2$ will be similar to that of $\max_{ii'} D_{ii'}^2$ for a fixed $\alpha$, although the resulting degrees of freedom might be different from that of $\max_{ii'} D_{ii'}^2$. If $\tau$ needs to be estimated, the null distribution of $\max_{ii'} D_{ii'}^2$ will be different from that of $\max_{ii'} D_{ii'}^2$. We leave this investigation to the future.

In Section 2.6.2, based on the assumption that the variances of the coherent clusters are homogeneous, we proposed to find a cut-value $h$ from a preliminary PfCluster dendrogram. However, this assumption might not be reasonable in some real data situations. A careful study of the impact of a proper variance stabilization method on PfCluster is needed. Developing a generalized procedure for finding $h$ based on an assumption of a multivariate variance-covariance (i.e allowing for some random effects) will also be interesting.

Enhancements for RP Algorithm

In our RPselection algorithm a $FIT$ measure is defined in Section 3.2.4 to select a set of most interesting genes. The distribution of $FIT^2$ is approximately $\chi^2$ when all profiles are parallel. The impact of incorporating a more accurate approximation to the distribution of the robust Mahalanobis distance will be investigated.

Currently the RPselection algorithm only works for categorical class variables defined on samples (samples). An extension to allow for continuous sample class variables may be possible using a regression tree idea in CART (Breiman, 1984). The key is to find a correspondence measure, $\phi$, that measures the concordance of clustering with respect to continuous sample characteristics.
Intuitively, the RPselection algorithm selects genes based on gene regulatory pathway. However, how the regulatory pathway is linked to the selection performance measure $J$ is unclear. More integration and a link between the RPselection algorithm and the genetic regulatory network research will be helpful to our understanding. Note that the RPselection algorithm is also applicable to SNP data analysis and other association or correlation based feature selection problems.

The RPselection algorithm currently works for up to 5000 features with a bootstrap size of 1 million. An algorithm with improved memory usage is needed for its application new microarray techniques that will generate data of increasing number of features. The RPselection algorithm is also CPU intensive. A better optimization may help to speed up the computation. It will be helpful to develop a portable parallel RPselection algorithm.

The idea of the RPselection algorithm may be adapted to semi-supervised learning. Some random walk approaches should also be applicable here.

There are other challenges and new research directions in data mining. Data mining has become one of the most important research areas in this modern information age.
Appendix

A.1 Permutation Matrix

A permutation matrix can be obtained by permuting the rows of an \( n \times n \) identity matrix according to a permutation of indices \( 1, \ldots, n \). Every row and column in a permutation matrix contain a single 1 and \( (n - 1) \) 0’s. A permutation matrix is of size \( n \) if it corresponds to a permutation of indices \( 1, \ldots, n \). There are total of \( n! \) permutations of size \( n \), with a one-to-one correspondence between each permutation and a permutation matrix.

A permutation matrix is nonsingular; its determinant is \( \pm 1 \). A permutation matrix \( \Pi \) is orthogonal, \( \Pi \Pi' = I \), (A.1)

where \( \Pi' \) is the transpose of \( \Pi \) and \( I \) is the identity matrix.

Suppose \( M \) is an arbitrary matrix, then

. \( \Pi M = M \) with rows interchanged by \( \Pi \),

. \( M \Pi = M \) with columns interchanged by \( \Pi \).

For example, the permutation matrices of size 2 are given by

\[
\begin{pmatrix}
1 & 0 \\
0 & 1
\end{pmatrix}, \quad \begin{pmatrix}
0 & 1 \\
1 & 0
\end{pmatrix}
\]
and of size 3 are given by

\[
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix},
\begin{pmatrix}
1 & 0 & 0 \\
0 & 0 & 1 \\
0 & 1 & 0
\end{pmatrix},
\begin{pmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
1 & 0 & 0
\end{pmatrix},
\begin{pmatrix}
0 & 1 & 0 \\
0 & 1 & 0 \\
1 & 0 & 0
\end{pmatrix},
\begin{pmatrix}
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0
\end{pmatrix},
\begin{pmatrix}
0 & 0 & 1 \\
0 & 1 & 0 \\
1 & 0 & 0
\end{pmatrix}.
\]

A.2 Lance and Williams Formula

The Lance and Williams (1967) recurrence formula (LW formula, hereafter) for a distance between a group \(k\) and a group \((ij)\) is

\[
d_{k(ij)} = \alpha_i d_{ki} + \alpha_j d_{kj} + \beta d_{ij} + \gamma |d_{ki} - d_{kj}|,
\]

where group \((ij)\) is formed by the fusion of two groups, \(i\) and \(j\), and \(d_{ij}\) is the distance between groups \(i\) and \(j\). Using Lance and Williams formula, LW developed a new ‘flexible’ scheme, with parameter values \(\alpha_i + \alpha_j + \beta = 1, \alpha_i = \alpha_j, \beta < 1, \gamma = 0\), to include all distance measures used in standard clustering methods.

The between-cluster distance measures used in standard hierarchical clustering techniques can be represented by the LW formula with a suitable choice of parameters \(\alpha_i, \alpha_j, \beta\) and \(\gamma\), see Table A.1. For example, the distance defined by the single linkage method corresponds to \(d_{k(ij)}\) with \(\alpha_i = \alpha_j = 1/2, \beta = 0, \gamma = -1/2\):

\[
d_{k(ij)} = \frac{1}{2} d_{ki} + \frac{1}{2} d_{kj} - \frac{1}{2} |d_{ki} - d_{kj}|. \tag{A.3}
\]

If \(d_{ki} > d_{kj}\), then \(|d_{ki} - d_{kj}| = d_{ki} - d_{kj}\) and \(d_{k(ij)} = d_{kj}\). If \(d_{ki} < d_{kj}\), then \(|d_{ki} - d_{kj}| = d_{kj} - d_{ki}\) and \(d_{k(ij)} = d_{ki}\), and hence the Lance and Williams
recurrence formula gives

\[ d_{k(ij)} = \min(d_{ki}, d_{kj}), \quad (A.4) \]

which is the between-cluster distance by the single linkage method.

<table>
<thead>
<tr>
<th>method</th>
<th>Parameters in Lance-Williams formula</th>
<th>( \alpha_i (\alpha_j = \alpha_i) )</th>
<th>( \beta )</th>
<th>( \gamma )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single linkage</td>
<td>1/2</td>
<td>0</td>
<td>-1/2</td>
<td></td>
</tr>
<tr>
<td>Complete linkage</td>
<td>1/2</td>
<td>0</td>
<td>1/2</td>
<td></td>
</tr>
<tr>
<td>Average linkage</td>
<td>( \frac{n_i}{n_i + n_j} )</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Centroid linkage</td>
<td>( \frac{n_i + n_j}{n_i + n_j} )</td>
<td>(-\frac{n_in_j}{(n_i + n_j)^2})</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Median linkage</td>
<td>1/2</td>
<td>-1/4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Ward’s method</td>
<td>( \frac{n_k + n_i + n_j}{n_k + n_i + n_j} )</td>
<td>(-n_k )</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Table A.1: Lance-Williams parameters for standard hierarchical clustering methods. \( n_k, n_i, n_j \) are the respective cluster sizes when between-cluster distance is for cluster \( k \) and cluster \( ij \). Cluster \( ij \) is formed by a fusion of clusters \( i \) and \( j \) (see \( (A.2) \)).

### A.3 Linear Sum Assignment Problem (LSAP)

The assignment problem is one of the fundamental combinatorial optimization problems. It has following general description:

> Given \( K \) agents and \( K \) tasks, suppose that any agent can be assigned to perform any task, which incurs a specific cost depends on which agent is assigned to the task. All tasks must be done and each must be by exactly one agent. What is the best assignment that minimizes the total cost?

The assignment problem is a special case of the transportation problem, which is a special case of the minimum cost flow problem, which in turn is a special case of a linear programming. While it is possible to solve any of these problems using the simplex algorithm, each particular problem can be solved by a
more efficient algorithm (than the simplex algorithm), which takes advantage of its special structure.

The Linear Sum Assignment Problem (LSAP) is the most famous assignment problem. Other kinds include the quadratic assignment problem and bottleneck assignment problem.

Next, we formulate the LSAP problem mathematically. Let \( C = (c_{ij}) \) be the \( k \times k \) cost matrix. Denote \( X = (x_{ij}) \) a binary assignment matrix such that

\[
x_{ij} = \begin{cases} 
1 & \text{if agent } i \text{ is assigned to task } j, \\
0 & \text{otherwise.}
\end{cases}
\]

Then, the LSAP is to find the best \( X \) to minimize

\[
\sum_{i=1}^{k} \sum_{j=1}^{k} c_{ij} x_{ij} \tag{A.5}
\]

under the constraint that \( x_{ij} = 0 \) or 1 and

\[
\sum_{j=1}^{k} x_{ij} = 1, \sum_{i=1}^{k} x_{ij} = 1 \quad i, j = 1, 2, \ldots, k \tag{A.6}
\]

The LSAP can be solved by linear programming using a Simplex-style primal algorithm as is done by R package \textit{lpSolve} (Buttrey, 2005). One can also transform the LSAP into a network flow problem, and use RELAX-IV to solve it, as is done in R package \textit{optmatch} (Hansen, 2005).
Bibliography


Allocco, D. J., Kohane, I. S., and Butte, A. J. (2004). Quantifying the


Lockhart, D. J., Dong, H., Byrne, M. C., Follettie, M. T., Gallo, M. V., Chee, M. S., Mittmann, M., Wang, C., Kobayashi, M., Horton, H., and


Yu, H., Yang, J., and Han, J. (2003). Classifying large data sets with hierarchical clusters. In *9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*.


