BIOMARKER DISCOVERY FOR MICROARRAY DATA BY ENRICHED METHODS, STOCHASTIC APPROXIMATION AND MIXED EFFECT MODELS

By

LAN YI

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Statistics

written under the direction of

Javier Cabrera

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

May, 2014

ABSTRACT OF THE DISSERTATION

Biomarker Discovery for Microarray Data by Enriched Methods, Stochastic Approximation and Mixed Effect Models

By LAN YI

Dissertation Director:

Javier Cabrera

Nowadays microarray technology enables scientists to monitor the expression levels of hundreds of thousands of genes simultaneously. Because of the high cost of such experiments, the sample size is small, typically, only a few dozen. In this thesis, we propose a new perspective on microarray data. We believe microarray data generally contain three types of signals: specific signal, non-specific signal and spurious signal. We propose an enriched method for biomarker discovery which strengthens the specific signal (biomarkers) and weakens the spurious signal. We show that our enriched version of principal component analysis will highlight the specific signals in the data and can help separate different signals. We also show that enriched principal component analysis along with linear discriminant analysis will improve the classification and prediction of microarray data, comparing to some other popular methods. The results from our method are easy to interpret, too. We also prove the stochastic approximation procedure used in conditional t test converges under some general assumptions. Finally we discuss about analyzing the data from one novel experiment to find groups of genes (biomarkers), applying hierarchical clustering and nonlinear mixed effect models.

Acknowledgments

I would like to express my sincerest gratitude

- to Dr. Javier Cabrera for supervising this thesis with great patience. He gave many great ideas about the topics covered in this thesis. I thank him for illustrating different methods, suggesting relative materials and sharing efficient algorithms to me. I feel blessed to have Dr. Cabrera as my advisor. I thank him for referring me to two summer internships in pharmaceutical companies, where I learned a lot about the nature of jobs in industrial areas and about social skills. He helped me not only on my research work but also on my personality development. I also thank him for helping me out in those messy situations because of my poor people skills.

- to Dr. Lee Dicker and Dr. Han Xiao for serving as my dissertation committee. I also learned a lot in Dr. Xiao's data mining course. It was a great experience.

- to Dr. Birol Emir for traveling a long way to be on the committee. I thank him for his advice and help when I was an intern in his team at Pfizer. I am also gratitude to him for forgiving my immature comments. I learned a lot about communication skills and social skills that summer. Thanks to his kindness, I had a great time.

- to Dr. Dhammika Amaratunga. He helped a lot on one topic in this thesis. Working with him at J&J, I learned a lot from him. Thanks to his great patience and kindness, I accomplished my first summer intern job successfully.

- to Dr. John Kolassa for helping me on choosing different courses and giving advices on a lot of things.

Dedication

To:

- Xianghua Yi: my dear father
- Shunfang Liu: my dear mother
- Lin Yi: my dear sister
- Guitang Lan: my dear boyfriend

I appreciate their understanding about my studying aboard and their encouragement and support about all my decisions. Without their love, I could not enjoy my life so freely.

Table of Contents

Chapter 1

Introduction

Nowadays microarray technology enables scientists to monitor the expression levels of hundreds of thousands of genes simultaneously. By studying the gene expression data, hopefully, we can get some idea about the biomarkers for the underlying biological state or condition. Because of the high cost of such experiments, the sample size is small, typically, only a few dozen. To analyze such high-dimensional but small-sample-size data, various methods were proposed.

One direction in microarray data analysis is to identify differentially expressed genes. For this purpose, different hypothesis testing procedures were suggested (we will review some of them in chapter 3). Most of these procedures analyze each gene separately. Classical two-sample t test is one natural choice. As it tends to give a high type I error rate for gene whose variability is low and a high type II error rate for gene whose variability is high, different ways were proposed to adjust classical two-sample t test for small-sample-size microarray data, for example, SAM t test, Limma t test and conditional t test. In addition, as we perform thousands of hypothesis testings simultaneously, how to control the overall Type I error is an issue. For example, suppose we are testing 10000 genes about whether each of them is differentially expressed. If we control each hypothesis test at level 0.01, assuming the 10000 tests are independent, we will identify 100 genes as differentially expressed wrongly on average, which is quite disturbing when the scientists are expecting to get around 500 differentially expressed genes in total. Traditionally, familywise error rate (section 3.2.1) is adopted to be the overall Type I error. However, controlling familywise error rate is too conservative for microarray data. Still using the example before, if we'd like to control the familywise error rate at level 0.05, then by Bonferroni method, the level of each test is only 5×10^{-6}. Thus we lose too much power. Benjamini and Hochberg

(1995) [9] proposed to control the false discovery rate, which will be discussed and used in this thesis whenever we need to deal with the overall Type I error of multiple hypothesis testings.

Another direction in microarray data analysis is to do classification and prediction, by which we could get insight ideas about the gene expression pattern of some underlying disease and help screening diseased tissues. Classical classification methods such as linear discriminant analysis require the sample size to be larger than the dimension of the data. Otherwise, the estimated within-class scatter matrix for microarray data is singular. Various methods were given to address the problem. Roughly speaking, these methods could be classified into two classes: one class of methods work on the matrix itself, manually making all eigenvalues positive; another class of methods try to reduce the high dimensionality of data by selecting a small subset of genes, or by projecting into a space with much smaller dimension, for example, applying principal component analysis or partial least squares. Statisticians argued that working with high-dimension matrix is not appealing as these estimated matrix is quite unstable when the sample size is small. They also argued that projecting into a small dimension space might give suboptimal answers. Some other classical classification methods, such as support vector machine and logistic regression, can be applied for classification and prediction of microarray data. They do not require the sample size to be larger than the dimension of the data. However, their performances are bad due to problems like overfitting. Adding penalties to their objective functions of these methods is suggested. Fast algorithms were written for these penalized methods to give efficient answers. However, the results from penalized methods are difficult to justify from biological view.

In this thesis, we propose a new perspective on microarray data. In our opinion, though its statistical structure could be quite complicated, microarray data generally contain three types of signals: specific signal, non-specific signal and spurious signal. Specific signal is generated by the process that the experiment is studying. Non-specific signal is due to some secondary effect. Spurious signal comes from the large proportion of noise, which is typically very strong in microarray data.

We propose an enriched method to strengthen the specific signal and weaken the spurious signal. We show that our enriched version of principal component analysis will highlight the specific signals in the data and can help separate different signals. We also show that enriched principal component analysis along with linear discriminant analysis will improve the classification and prediction of microarray data, comparing to some other popular methods. By checking the loadings of the enriched principal components, we can find a set of genes which will help the scientists to find the biomarkers. The results from our method are easy to interpret, too.

To address the small sample size problem, conditional t test is applied in the weighting procedure. In this thesis, we also try to show that the stochastic approximation procedure used in conditional t test converges under some general assumptions. We will show that the stochastic approximation algorithm can be applied to improve the performance of the sample correlation matrix.

With the development of microarray technology, new ideas about adopting microarray technology to study the behavior of each gene are proposed. We will introduce some new technology in this thesis, which can be used to shut down one specific gene. We will discuss our way to analyze such data to find out a set of interesting genes (biomarkers), applying hierarchical clustering and nonlinear mixed effect methods.

The rest of this thesis is organized as follows. In chapter 2, we will review the concepts of gene and gene expression, describe a typical microarray experiment, give a rough idea about how to deal with the scanned image to get intensity values, and outline the preprocessing steps. In chapter 3, we will review different methods for identifying differentially expressed genes, discuss the familywise error rate and false discovery rate and review different methods for classification and prediction of microarray data. In chapter 4, we will discuss the details about our enriched principal component analysis and linear discriminant analysis methods and show the advantages of our method by one microarray data. In chapter 5, we will prove that the stochastic approximation procedure used in conditional t test converges under some general assumptions and show the power of the procedure by simulation. In chapter 6, data analysis about one novel experiment is discussed. Finally in chapter 7, we give a short summary about the topics in this thesis and discuss future research.

Chapter 2

Preliminaries

2.1 Gene and Gene Expression

A gene is a stretch of deoxyribonucleic acids (DNA) that has a function in the organism, holding information to build and maintain an organism's cells and passing genetic traits to offspring. A DNA molecule consists of two long strands wound tightly around each other in a spiral structure known as a double helix, which is chemically inert. Hence, DNA is a stable carrier of genetic information. Each strand of the DNA molecule is a linear arrangement of repeating similar units called nucleotides, which composed of a carbon sugar (deoxyribose), a phosphate group and a nitrogenous base. The fundamental component of each nucleotide is a nitrogenous base, which could be adenine (A), thymine (T), guanine (G) or cytosine (C). The pairing rules of these four nitrogenous bases are: guanine only pairs with cytosine; adenine only pairs with thymine (this rule is called 'the complementary base-pairing rules'). Therefore, the two strands of DNA must be complementary. Due to the chemical composition of the pentose residues of the bases, DNA strands have directionality. All nucleic acid synthesis in a cell occurs in one particular direction. DNA is organized into long structures (called chromosomes) in cells. The total complement of genes in an organism or cell is known as its genome, which may be stored on one or more chromosomes.

There is another type of nucleic acid which performs multiple vital roles in the coding, decoding, regulation and expression of genes: ribonucleic acid (RNA). RNA is a single-stranded chain of nucleotides, which contains a ribose sugar instead of a deoxyribose sugar as in DNA. In addition, RNA contains uracil instead of thymine.

2.1.1 Gene Expression

Gene expression is the process by which information from a gene is used in the synthesis of a functional gene product. Generally, the product of the gene expression process is some protein, which contains one or more chains of amino acids. Proteins differ from one another primarily in their sequence of amino acids, which is dictated by the nucleotide sequence of their genes. Genes that encode proteins are composed of a series of three-nucleotide sequences (codons). Codons serve as the words in the genetic language. The genetic code specifies the correspondence during protein translation between codons and amino acids. Proteins are large biological molecules, or macromolecules. They perform and regulate most of life's basic functions, including catalyzing metabolic reactions, replicating DNA, responding to stimuli, and transporting molecules from one location to another.

There are several steps in the gene expression process: transcription, RNA processing, non-coding RNA maturation, RNA export, translation, folding, translocation, and protein transport. In our study, we focus on the transcription and translation steps.

In transcription step, gene information must be transcribed from DNA to messenger RNA (mRNA). The nucleotide sequence of mRNA is complementary to the DNA from which it is transcribed. The DNA strand whose sequence matches that of the RNA is known as the coding strand and the strand from which the RNA is synthesized is the template strand. In translation step, information is translated from mRNA into protein. Translation of the mRNA requires RNA adaptor molecules called tRNAs. Each triplet codon is recognized by a tRNA, which is associated with a cognate amino acid. Out of 64 ($4 \times 4 \times 4$) potential codons, 61 are used to specify the 20 amino acid building blocks of proteins, whereas 3 are used to provide chain-terminating signals (UAA, UAG, UGA).

2.2 Microarray

Knowing which subset of genes are expressed in one particular cell at a given time, we could make inferences about the state of the cell. By studying the mRNAs (the product in the transcription step of gene expression) or proteins (the product in the translation step), we will get some insight about the subset of expressed genes. Comparing to mRNA,

studying proteins could be more complicated. For one thing, the function of a protein is determined by not only the amino acid sequence but also its spacial structure; for another, it is more difficult to purify proteins. Therefore, scientists adopted the way of studying the type and the quantities of mRNAs presented in one cell. The complete collection of mRNAs (including their alternative splicing variants) is referred to as the organism's transcriptome.

The DNA microarray is the most widely used technology for studying gene expression levels. This technology lies on the hybridization fact: two DNA strands (or one DNA strand and one mRNA strand) will hybridize with each other, regardless of whether they originated from a single source or from two different sources, as long as their base pair sequences match according to the complementary base-pairing rules. A hybrid DNA molecule will be formed as long as there is sufficient similarity between the two strands.

2.2.1 A Typical Microarray Experiment

There are four basic steps in a typical microarray experiment (this part is taken from Amaratunga and Cabrera (2003) [1]).

Microarray Preparation

A DNA microarray (also commonly known as DNA chip or biochip) is a collection of microscopic DNA spots attached to a solid surface. A drop of each type of purified single-stranded DNAs in some collection is placed onto a specially prepared glass microscope slide by a robotic machine called an arrayer. This process is called arraying or spotting. The arrayer can quickly produce a regular grid of thousands of spots in a dime-sized area. The DNA in the spots is bonded to the glass to keep it from washing off during the hybridization reaction and subsequent wash. If it is cDNA used in the arraying procedure, the microarray is called cDNA microarray; if it is oligonucleotides, the microarray is called oligonucleotide array.

Sample Preparation

The sample is prepared by purifying mRNA from total cellular contents. As mRNA degrades quickly, it is reverse-transcribed into more stable cDNA or cRNA. Fluorescent dyes (which

will fluoresce when exposed to a specific wavelength of light) are used to label the sample so that it is able to detect which cDNAs are bound to the microarray. We call a microarray with only one fluorophore one-channel microarray. When two samples are applied to the same microarray (for example, one sample from diseased tissue and the other from healthy tissue), two different fluorophores are applied (for example, the diseased sample is labeled with green fluorophore and the healthy one with red fluorophore). We call such a microarray two-channel microarray.

The Hybridization Step

The labeled sample is poured onto the microarray at this step. The scientists will make sure that the sample diffuses uniformly all over the microarray. Then it is sealed in a hybridization chamber and incubated at a specific temperature for enough time to ensure that the hybridization reactions complete. All areas of the microarray should be exposed to a uniform amount of labeled sample all the time. The microarray is then removed from the hybridization chamber and thoroughly, but carefully, washed to eliminate any excess labeled sample. Finally the microarray is dried up.

Scanning the Microarray

When hybridization is completed, the microarray is scanned to determine the amount of labeled sample bound to each spot. The emitted light from the fluorophore when stimulated by a laser is captured by a scanner and the intensity is recorded. Spots with more bound sample will have larger intensities. The result is a series of images.The scanner "reads" a microarray by dividing it up 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.

2.3 Processing the Scanned Image

The image of microarray must be converted from pixel intensities into spot intensities, so that every DNA sequence that was spotted on the microarray gets an intensity measure,

called the spot intensity, reflecting the amount of labeled sample that hybridized to it.

2.3.1 Converting the Scanned Image to the Spotted Image

Gridding

As the arraying process in practice is not perfect, the grid that is actually arrayed is not a regular rectangular grid. To define the center of the spot, we overlay an appropriately sized grid on the microarray and then manipulate the rows and columns to align more properly.

Segmentation

Because the spots vary considerably in size, shape and regularity, the region of the slide on which cDNA was arrayed (called the spot) needs to be separated from the background. One seeded region growing algorithm (Amaratunga and Cabrera (2003) [1] was suggested for this problem, which allocates the pixels to either signal or background region.

Quantification

Quantification step will assign each spot an intensity value.

Spot Intensity The intensity of the spot located at the rth row and cth column of the array will be denoted as $\{SI_{rc}\}$, which is the average intensity of the pixels in that spot which are designated as signal.

Spot background The intensity of the background for the spot located at the rth row and cth column of the array will be denoted as $\{BI_{rc}\}$, which is the average intensity of the pixels around the spot which are designated as background.

2.3.2 Quality Assessment

Once the spotted image and related statistics are obtained, it is advisable to (1) assess the quality of the array and (2) evaluate the quality of the individual spots on the array. For example, checking whether the background intensities $\{BI_{rc}\}$ are uniformly distributed; or check whether the extreme values in either the spot or the background are randomly scattered throughout the array or clustered together or distributed according to some pattern.

2.3.3 Adjusting for Background

In reality, the background intensity from the image data is not zero because of nonspecific fluorescence. It is concerned that the raw spot intensities may also contain some amount of the nonspecific fluorescence. Therefore, the raw spot intensities should be adjusted for background. It is assumed that the spot signal intensity is an additive combination of the true spot intensity and the background.

There are several ways to estimate background: global background adjustment (by the average intensity of all the pixels not belonging to spots), spot background adjustment (by the spot background intensity), smoothed background adjustment (running a simple smoothing procedure through the array) and zonal background adjustment (a variation of smoothed background adjustment)

Suppose that the spot intensity at the gth spot is SI_g and the background intensity is estimated to be BI_g. The background-adjusted spot intensity value, AI_g, is obtained by shifting the spot intensity down by the background intensity:

$$AI_g = SI_g - BI_g.$$

To make sure AI_g is positive, a threshold is set. For example, if T is a low percentile of the SI_g values (say, the fifth percentile), take the background-adjusted thresholded spot intensity value, AI_g, to be

$$AI_g = \max(SI_g - BI_g, T).$$

2.4 Preprocessing Microarray Data

Preprocessing prior to formal analysis is needed to address several data-related issues: to transform the data into a scale suitable for analysis; to remove the effects of systematic sources of variation; and to identify discrepant observations and arrays.

2.4.1 Logarithmic Transformation

It is preferable to work with logged intensities rather than absolute intensities for a number of reasons: the variation of logged intensities tends to be less dependent on the magnitude of

the values; taking logs reduces the skewness of highly skewed distributions; and taking logs improves variance estimation. Moreover logged intensities facilitate visual inspection of the data: the data is spread out more evenly. For some other variance stabilizing transformation, please check Amaratunga and Cabrera (2003) [1].

2.4.2 Normalization

It has been noticed that substantial differences in intensity measurements exist even among microarrays that were treated exactly alike. The differences can generally be traced to systematic effects. To remove these effects and improve the comparability among microarrays which are treated alike, normalization process was introduced.

One popular method is quantile normalization, whose objective is to make the distributions of the transformed spot intensities, as similar as possible across the microarrays. Either a subset of quantiles or all the quantiles may be equated. For the details about different normalization methods, please check Amaratunga and Cabrera (2003) [1].

In the following chapters, we assume our microarray data has already been suitably transformed and normalized.

2.5 Replicates

2.5.1 Technical Replicates

Technical replicates are used to deal with technical variation, which arises from the handling steps, such as mRNA extraction, amplification, labeling, hybridization, and scanning. This variation introduces uncertainty to the intensity measurements associated with a gene. Using technical replicates and averaging across them allows gene expression levels to be estimated with greater precision. The higher the number of replicates, the greater the precision.

2.5.2 Biological Replicates

Biological replicates are used to deal with biological variation, which is the natural variability among subjects due to genetic diversity, environmental effects and other causes. This

variation also contributes uncertainty to the intensity measurements associated with a gene. Using biological replicates and averaging across them allows gene expression levels to be estimated with greater biological precision. The higher the number of replicates, the greater the precision.

Chapter 3

Microarray Data Analysis

3.1 Analyze each Gene Separately

Gene expression analysis across various biological conditions, cell cycle states, tissues and subjects may help identify differentially expressed genes, which could be interesting biomarkers. This type of information is a valuable pinpoint in the investigation of biological processes and functional disorders. The idea of most of the statistical methods for identifying differentially-expressed genes is to do hypothesis testing for each gene with null hypothesis 'H_0 : gene g is not differentially-expressed', though different methods would choose different test statistics.

The following notation will be used in this chapter. Assume we are comparing the expression levels of a set of G genes in two groups of microarrays: Group 1 and Group 2. There are n_1 microarrays in Group 1 and n_2 microarrays in Group 2, and the total sample size is $N = n_1 + n_2$. Let x_{gij} denote the intensity measurement for the gth gene in the ith microarray in the jth group, where $i = 1, \cdots, n_j;\ j = 1, 2$; and $g = 1, \cdots, G$. Let $\overline{x}_{gj}, s_{gj}$, denote, respectively, the mean and standard deviation of gene g in the jth group.

3.1.1 Basics of Statistical Hypothesis Testing

With microarray data, there are G null hypotheses being tested, the gth null hypotheses, for $g = 1, \cdots, G$, being that the gth gene is not differentially expressed across the groups. If the decision of the test is to reject the null hypothesis, we get a positive finding. The decision can be true or false:

True Positive: if the null hypothesis is false;

False Positive: if the null hypothesis is true; this is also called a Type I error.

If the decision of the test is not to reject the null hypothesis, we get a negative finding. The decision can be true or false:

True Negative: if the null hypothesis is true;

False Negative: if the null hypothesis is false; this is also called a Type II error.

3.1.2 Fold Changes

Early analyses of microarray data declared a gene differentially expressed if its fold increase or fold decrease exceeded a specified cutoff. On a logarithmic scale, the decision rule is: if

$$|\overline{x}_{g2} - \overline{x}_{g1}| > \log(h),$$

we say gene g is differentially-expressed. In other words, when the change is larger than h-fold, the gene is differentially-expressed.

Schena et al. (1995) [58] declared a gene differentially expressed if its expression level showed a fivefold difference between the two mRNA samples. Using fold change has been criticized, since genes with high variability have a reasonable probability of having a large fold change which means the fold change could be statistically non-significant.

3.1.3 The Two-Sample t Test

The more basic statistical test for comparing the means of two groups is the two-sample t test. The two-sample t test statistic is given by

$$T_{ge} = \frac{|\overline{x}_{g1} - \overline{x}_{g2}|}{s_{gp}\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}},$$

where

$$s_{gp}^2 = \frac{(n_1 - 1)s_{g1}^2 + (n_2 - 1)s_{g2}^2}{n_1 + n_2 - 2} \tag{3.1}$$

is the pooled estimate of variance.

If the data is drawn from a normal distribution and is homoscedastic: $x_{gij} \sim N(\mu_{gj}, \sigma_g^2)$, the null distribution of T_{ge} is a t-distribution with degrees of freedom $\nu = n_1 + n_2 - 2$. If

the observed value of T_{ge} is $T_{ge;obs}$, gene g is declared significantly differentially expressed at level of significance α if $p_{ge} = \Pr(|T_{ge}| > T_{ge;obs}) < \alpha$.

There are times we would like to focus on more meaningful differences - the absolute difference of the average intensities in the two groups is larger than a specified value Δ ($\Delta = \log(2)$ for a twofold difference):

$$T_{g\Delta} = \frac{|\overline{x}_{g1} - \overline{x}_{g2}| - \Delta}{s_{gp}\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}.$$

The null distribution of $T_{g\Delta}$ is a t-distribution with degrees of freedom $\nu = n_1 + n_2 - 2$.

The assumptions of normality and homoscedasticity are critical for the t test to function properly. When the assumption of homoscedasticity is not tenable, Welch's t test is proposed:

$$T_{gu} = \frac{|\overline{x}_{g1} - \overline{x}_{g2}| - \Delta}{\sqrt{\frac{s_{g1}^2}{n_1} + \frac{s_{g2}^2}{n_2}}}.$$

The null distribution of T_{gu} is approximately a t-distribution with degrees of freedom:

$$\nu_g = \frac{\left(\frac{s_{g1}^2}{n_1} + \frac{s_{g2}^2}{n_2}\right)^2}{\frac{1}{n_1 - 1}\left(\frac{s_{g1}^2}{n_1}\right)^2 + \frac{1}{n_2 - 1}\left(\frac{s_{g2}^2}{n_2}\right)^2}.$$

If the observed value of T_{gu} is $T_{gu;obs}$, gene g is declared significantly differentially expressed at level of significance α if $p_{gu} = \Pr(|T_{gu}| > T_{gu;obs}) < \alpha$.

For microarray data, the small sample size makes the t test unreliable: a high false positive rate for genes whose variability is low and a high false negative rate for genes whose variability is high.

3.1.4 Small Variance-Adjusted t Tests: SAM t Test

One method to adjust t tests for microarray data was suggested by Tusher et al. (2001) [71]. They added a carefully chosen constant c to the denominator of the t statistic:

$$T_{gs}(c) = \frac{|\overline{x}_{g1} - \overline{x}_{g2}|}{s_{gp}\sqrt{\frac{1}{n_1} + \frac{1}{n_2}} + c},$$

where s_{gp} is defined at (3.1). This test statistic is often called the SAM t statistic, where SAM stands for "significance analysis of microarrays".

Computation of c

1. Let s^α be the α percentile of the s_{gp} values.
2. Compute the 100 quantiles of the s_{gp} values, denoted by $q_1 < q_2 < \cdots < q_{100}$.
3. For $\alpha \in (0, 0.05, 0.10, \cdots, 1.0)$, compute ν_j =mad$(T_{gs}(s^\alpha)|s_{gp} \in [q_j, q_{j+1}))$, $j = 1, 2, \cdots, 100$, where mad is the median absolute deviation from the median, divided by 0.64. Compute cv(α) = coefficient of variation of the ν_j values.
4. Choose $\hat{\alpha} = \arg\min[\text{cv}(\alpha)]$. Then $c = s^{\hat{\alpha}}$.

Assess Significance

1. Compute order statistics $T_{(1)s}(c) \leq T_{(2)s}(c) \leq \cdots \leq T_{(G)s}(c)$.
2. Take B sets of permutations of the samples. A balanced permutation will be each of the new groups contain $n_1/2$ Group 1 samples and $n_2/2$ Group 2 samples. For each permutation b compute statistics $T^{\star b}_{gs}(c)$ and corresponding order statistics $T^{\star b}_{(1)s}(c) \leq T^{\star b}_{(2)s}(c) \leq \cdots \leq T^{\star b}_{(G)s}(c)$.
3. For the set of B permutations, estimate the expected order statistics by $\overline{T}_{(g)s}(c) = \sum_b T^{\star b}_{(g)s}(c)/B$ for $g = 1, 2, \cdots, G$.
4. Plot the $T_{(g)s}(c)$ values versus the $\overline{T}_{(g)s}(c)$.
5. For a fixed threshold Δ, starting at the origin, and moving up to the right find the first $g = g_1$ such that $|T_{(g)s}(c) - \overline{T}_{(g)s}(c)| > \Delta$. All genes past g_1 are called "significance".

3.1.5 Conditional t Test

By assuming (a) σ_g is the same for both groups and is distributed as F_σ and (b) σ_g is independent of μ_g for both groups, Amaratunga and Cabrera (2009) [2] proposed a novel method to address the dependence of T_{ge} from s_{gp} by determining, from the distribution of T_{ge} conditioned on s_{gp}, the critical value of T_{ge} that separates significance from nonsignificance. They borrowed information from other genes in estimating σ_g's distribution F_σ. This method is called the conditional t (CT) approach, containing the following two steps.

Step 1. Estimate F_σ

Due to the small sample size of microarray data, the empirical distribution of s_g, $\hat{F}_s$, is a biased estimator of the distribution F_σ. The bias can be corrected by using stochastic approximation method to solve the target estimation (Cabrera and Fernholz (1999) [21] and Cabrera and Watson (1997) [24]) of $\hat{F}_s$ through bootstrap. The idea is to estimate the function $g : [0,1] \to [0,1]$ defined by $g(F_\sigma(x)) = \hat{F}_s(x)$. Then the bias-corrected estimate of F_σ will be $\hat{g}^{-1}(\hat{F}_s)$. The algorithm is as follows:

1. Generate a null distribution for the data by subtracting the sample means and dividing by the standard deviations.

2. Assume that $\hat{F}_s(x)$ is the true distribution of σ. Then sample from the null distribution of x to get a sample of size N and multiply the sample by a σ generated from $\hat{F}_s(x)$. Repeat this 10,000 times and get 10,000 sets of samples.

3. For each set of samples, calculate a value for the pooled sample standard deviation, namely $s_b^\star$, for $b = 1, \cdots, 10,000$. Let $\hat{F}_{s^\star}(x)$ be the empirical distribution of the $s_b^\star$'s. Then the estimator of g is obtained by mapping the empirical distribution $\hat{F}_s$ into $\hat{F}_{s^\star}$:

$$\hat{g}(y = \hat{F}_s(x)) = \hat{F}_{s^\star}(\hat{F}_s^{-1}(y))$$

and

$$\hat{g}^{-1}(y) = \hat{F}_s(\hat{F}_{s^\star}^{-1}(y)).$$

Hence the bias corrected estimator of F_σ is

$$\hat{F}_\sigma(x) = \hat{F}_s(\hat{F}_{s^\star}^{-1}(\hat{F}_s(x))).$$

$\hat{F}_\sigma(x)$ will be used in the second part of the method to generate the standard deviations of the gene populations.

Step 2. Estimate the Conditional Distribution of $T_g|s_g$

1. Generate a null distribution for the data by subtracting the sample means and dividing by the standard deviations.

2. Resample from the null distribution of x and multiply each sample by a σ generated from $\hat{F}_\sigma(x)$. Repeat this 10,000 times and obtain 10,000 sets of samples. From each set of samples, calculate a value for the pooled sample standard deviation and the two-sample t statistic, namely s_g and t_g for $g = 1, \cdots, 10,000$.

3. Estimate $t_\alpha(s_g)$ using a quantile regression estimate for t_g versus s_g and estimate the regression quantile curve for the $1-\alpha$ quantile.

Denote the pooled variance estimate as s and let t be the statistic for a randomly selected gene. Let $f(t,s)$ be the joint probability density function of t and s. The CT procedure rejects the null hypothesis if $t > h(s)$ and conditioning on s the probability of type I error is α. Then the overall unconditional probability of type I error is also α:

$$\begin{aligned}\int_0^\infty \int_{h(s)}^\infty f(t,s)dtds &= \int_0^\infty \left(\int_{-\infty}^\infty f(t,s)dt\right) \frac{\int_{h(s)}^\infty f(t,s)dt}{\int_{-\infty}^\infty f(t,s)dt} ds \\ &= \int_0^\infty \int_{-\infty}^\infty f(t,s)\alpha dtds = \alpha.\end{aligned}$$

3.1.6 LIMMA t Test

Smyth (2004) [61] built hierarchical model to "borrow strength across genes" by assuming prior distributions for the sets of parameters. They assumed

$$\begin{aligned}\overline{x}_{g1} - \overline{x}_{g2} | \mu_{g1} - \mu_{g2}, \sigma_g^2 &\sim N(\mu_{g1} - \mu_{g2}, \nu_g \sigma_g^2) \\ s_g^2 | \sigma_g^2 &\sim \frac{\sigma_g^2}{d_g} \chi^2_{d_g} \\ \frac{1}{\sigma_g^2} &\sim \frac{1}{d_0 s_0^2} \chi^2_{d_0} \\ \Pr(\mu_{g1} - \mu_{g2} \neq 0) &= p \\ \mu_{g1} - \mu_{g2} | \sigma_g^2, \mu_{g1} - \mu_{g2} \neq 0 &\sim N(0, \nu_0 \sigma_g^2)\end{aligned}$$

They used the posterior mean of σ_g^2 ($\widetilde{s}_g^2$) in place of ordinary sample standard deviations to moderate t-test

$$\widetilde{t}_g = \frac{\overline{x}_{g1} - \overline{x}_{g2}}{\widetilde{s}_g \sqrt{\nu_g}},$$

where $\widetilde{s}_g^2 = E(\sigma_g^2 | s_g^2) = \frac{d_0 s_0^2 + d_g s_g^2}{d_0 + d_g}$. And they showed that under null hypothesis:

$$\widetilde{t}_g \sim t_{d_0 + d_g}.$$

3.2 Multiplicity

In identifying differentially-expressed genes, we perform thousands of hypothesis testings simultaneously. For these multiple inferences, we would like to calculate and control the overall type I error, as unguarded use of single-inference procedures will increase false positive rate. For example, suppose we are testing about 5000 genes, if we treat these tests independent, and set the significant level of each one as 0.05, on average, we could falsely identify 250 genes as differentially-expressed, which could be quite misleading.

There are several ways of controlling the overall type I error rate. Before we discuss them, let's introduce some notations. Consider testing simultaneously G (null) hypotheses, of which m_0 (unknown) are true. R is the number of hypotheses rejected, which is observable. But U, V, S, T are unobservable random variables.

	Declared non-significant	Declared significant	Total
True null hypotheses	U	V	m_0
Non-true null hypotheses	T	S	$G - m_0$
	$G - R$	R	G

Table 3.1: Multiple Hypothesis Testing

3.2.1 Familywise Error Rate (FWER)

The traditional way to control the overall type I error rate is to control

$$P(V \geq 1),$$

that is, controlling the probability of committing any false positive among all the hypotheses test, which is called familywise error rate (FWER). Several simple and sequential multiplicity adjustment methods for FWER will be reviewed here.

Suppose $p_1, \cdots, p_G$ are the G observed p-values for G statistical tests. Classical p-value adjustments are single-step procedures in that the same adjustment is applied to each p-value regardless of their ordering.

- Bonferroni: The Bonferroni p-value for the kth test is simply $\widetilde{p}_k^B = Gp_k$. If it exceeds 1, it is set to 1. The Bonferroni adjustment is highly conservative, consequently the

adjusted tests have low power.

- Sidak: The Sidak p-value for the kth test is $\widetilde{p}_k^S = 1 - (1 - p_k)^G$. Sidak p-values are slightly less conservative than Bonferroni p-values.

An alternative approach is sequential p-value adjustment, which takes the order of the observed p-values into account with smaller p-values being adjusted more than larger p-values. Suppose that the unadjusted p-values have been ordered so that: $p_{(1)} < p_{(2)} < \cdots < p_{(G)}$.

- Holm-Bonferroni (Holm (1979)[44]): The Holm-Bonferroni step-down p-values are determined as

$$\begin{aligned}
\widetilde{p}_{(1)} &= Gp_{(1)} \\
\widetilde{p}_{(2)} &= \max(\widetilde{p}_{(1)}, (G-1)p_{(2)}) \qquad (3.2) \\
&\vdots \\
\widetilde{p}_{(j)} &= \max(\widetilde{p}_{(j-1)}, (G-j+1)p_{(j)}) \qquad (3.3) \\
&\vdots \\
\widetilde{p}_{(G)} &= \max(\widetilde{p}_{(G-1)}, p_{(G)})
\end{aligned}$$

  As always, if any adjusted p-value exceeds 1, it is set to 1.

- Holm-Sidak: The Holm-Sidak step-down p-values are determined similarly as Holm-Bonferroni procedures: taking the adjustments to be $1 - (1 - p_{(2)})^{(G-1)}$ at (3.2) and $1 - (1 - p_{(j)})^{(G-j+1)}$ at (3.3).

- Hochberg: Assuming that the G p-values are independent under their respective null

hypotheses, Hochberg (1988) [43] provided a set of step-up p-values:

$$\begin{aligned}
\widetilde{p}_{(G)} &= p_{(G)} \\
\widetilde{p}_{(G-1)} &= \min(\widetilde{p}_{(G)}, 2p_{(G-1)}) \\
&\vdots \\
\widetilde{p}_{(j)} &= \min(\widetilde{p}_{(j+1)}, (G-j+1)p_{(j)}) \\
&\vdots \\
\widetilde{p}_{(1)} &= \min(\widetilde{p}_{(2)}, Gp_{(1)})
\end{aligned}$$

He showed that this procedure is sharper than Holm-Bonferroni step down procedure under the assumption of independence.

- Westfall-Young (Westfall and Yong (1993) [73]): Let the ordered p-values have indexes $r_1, r_2, \cdots, r_G$, so that $p_{(1)} = p_{r_1}, p_{(2)} = p_{r_2}, \cdots, p_{(G)} = p_{(r_G)}$. Denote $H_0 = \cap_{i=1}^{G} H_i$. The Westfall-Young step-down p-values are determined as

$$\begin{aligned}
\widetilde{p}_{(1)} &= \Pr(\min_{l \in \{r_1, r_2, \cdots, r_G\}} P_l \leq p_{(1)} | H_0) \\
\widetilde{p}_{(2)} &= \max[\widetilde{p}_{(1)}, \Pr(\min_{l \in \{r_2, \cdots, r_G\}} P_l \leq p_{(2)} | H_0)] \\
&\vdots \\
\widetilde{p}_{(j)} &= \max[\widetilde{p}_{(j-1)}, \Pr(\min_{l \in \{r_j, \cdots, r_G\}} P_l \leq p_{(j)} | H_0)] \\
&\vdots \\
\widetilde{p}_{(G)} &= \max[\widetilde{p}_{(G-1)}, \Pr(P_{r_G} \leq p_{(G)} | H_0)]
\end{aligned}$$

These adjusted p-values are estimated by simulation and can be computationally intensive. But as this method takes into account the dependence characteristics among the tests, it is less conservative.

The restrictiveness of the familywise error rate criterion makes multiple testing procedures not powerful in the sense that the probability of rejecting null hypotheses that are false will be small.

3.2.2 False Discovery Rate

As controlling FWER is quite conservative for microarray data analysis, more allowance needs to be given. Benjamini and Hochberg (1995) [9] suggested controlling false discovery rate (FDR) instead. The FDR is defined as the expected proportion of false positives among the positive findings (the expected proportion of true null hypotheses which are erroneously rejected, out of the total number of hypotheses rejected):

$$\text{FDR} = E(\frac{V}{R}) = E[\frac{V}{R}|R > 0]\,\Pr(R > 0).$$

Controlling false discovery rate allows investigators to increase power while maintaining a principled bound on error. If all the null hypotheses are true (i.e., $m_0 = G$), the FDR is equal to the FWER and controlling the FDR would be equivalent to controlling the FWER. If not every null hypothesis is true (i.e., $m_0 < G$), the FDR is smaller than or equal to the FWER. Hence, any procedure controlling the FWER also controls the FDR. A procedure controlling the FDR can be less stringent and give more power. The potential for increase in power is larger when more hypotheses are not true. Benjamini and Hochberg (1995) [9], also Yekutieli and Benjamini (1999) [76], suggested the following step-up procedure to adjust the ordered p-values in order to control the FDR when the test statistics are independent:

- Benjamini-Hochberg: The Benjamini-Hochberg adjusted p-values are

$$\begin{aligned} \widetilde{p}^{BH}_{(G)} &= p_{(G)}, \\ \widetilde{p}^{BH}_{(G-k)} &= \min\left(\widetilde{p}^{BH}_{(G-k+1)}, \frac{G}{G-k}p_{(G-k)}\right), \end{aligned} \tag{3.4}$$

Benjamini and Yekutieli (2001) [13] shows that the above procedure controls FDR such that FDR $\leq q \cdot m_0/G$ for positively dependent test statistics. Depending on m_0/G, the Benjamini and Hochberg procedure could be too conservative. More power could be gained if we could estimate m_0 and use $\frac{\hat{m_0}}{G-k}$ in (3.4). Benjamini and Hochberg (2000) [10] and Benjamini et al. (2006) [11] implemented the idea by some adaptive two-step procedures.

The idea used in Westfall and Young (1993) [73] was applied in Yekutieli and Benjamini (1999) [76] to propose a resampling FDR adjustments to account for the dependence structure between the test statistics. Yekutieli (2008) [75] presented a modification of the

Benjamini and Hochberg false discovery rate controlling procedure for testing non-positive dependent test statistics. Romano et al. (2008) [56] applied bootstrap and subsampling to control FDR under dependence.

There are many other articles about Benjamini and Hochberg (1995) [9] FDR control procedure. For example, Reiner et al. (2003) [54] studied some of these procedures using simulation and found that they controlled FDR at the desired level. Benjamini and Liu (1999) [12] proposed a set of step-down adjusted p values: when $i \leq m(1-q)+1$,

$$\widetilde{p}_{(i)} = \frac{m-i+1}{m}\left[1-(1-p_{(i)})^{m-i+1}\right];$$

when $i > m(1-q)+1$,

$$\widetilde{p}_{(i)} = 1.$$

The step-down procedure neither dominates nor is dominated by the step-up procedure. In a large simulation study of the power of the two procedures, the step-down procedure turns out to be more powerful when the number of test hypotheses is small and many of the hypotheses are far from being true. Genovese et al. (2006) [36] suggested incorporating prior information about the hypotheses when controlling FDR. Yekutieli (2012) [77] proposed hierarchical FDR-controlling methodology.

3.2.3 The Positive False Discovery Rate

Storey (2003) [63] defined the positive false discovery rate (pFDR)

$$\text{pFDR} = E\left[\frac{V}{R}|R>0\right],$$

emphasizing the fact that we only worry about false discovery rate when there are rejections. By the definition, pFDR can only be estimated for a fixed significance region (Γ):

$$\text{pFDR}(\Gamma) = E\left[\frac{V(\Gamma)}{R(\Gamma)}|R(\Gamma)>0\right].$$

Positive false discovery rate has a good interpretation under Bayesian principals. Let $H_i = 0$ when the ith null hypothesis is true and $H_i = 1$ when it is false, $i = 1, \cdots, G$. Denote the statistics used for these tests as $T_1, \cdots, T_G$. When assuming that (T_i, H_i) are i.i.d. random variables, $T_i|H_i \sim (1-H_i)\cdot F_0 + H_i \cdot F_1$ for some null distribution F_0 and alternative

distribution F_1, and $H_i \sim \text{Bernoulli}(\pi_1)$ for $i = 1, \cdots, G$, the positive false discovery rate is the following posterior probability:

$$\text{pFDR}(\Gamma) = \Pr(H = 0 | T \in \Gamma),$$

where $\pi_0 = 1 - \pi_1$. In addition, under these assumptions, the following equation holds:

$$E\left[\frac{V(\Gamma)}{R(\Gamma)} | R(\Gamma) > 0\right] = \frac{E[V(\Gamma)]}{E[R(\Gamma)]}.$$

Storey (2002) [62] proposed one approach to estimate the pFDR on the basis of independent p-values: all rejection regions are of the form $p \in [0, \gamma]$ for some $\gamma \geq 0$.

$$\begin{aligned} \text{pFDR}(\gamma) &= \Pr(H = 0 | P \leq \gamma) \\ &= \frac{\pi_0 \Pr(P \leq \gamma | H = 0)}{\Pr(P \leq \gamma)} \\ &= \frac{\pi_0 \gamma}{\Pr(P \leq \gamma)}. \end{aligned}$$

A conservative estiamte of π_0 is

$$\hat{\pi}_0(\lambda) = \frac{\#\{p_i > \lambda\}}{(1-\lambda)G} = \frac{W(\lambda)}{(1-\lambda)G}$$

for some well-chosen λ, where $p_1, \cdots, p_G$ are the observed p-values and $W(\lambda) = \#\{p_i > \lambda\}$. A natural estimate of $\Pr(P \leq \gamma)$ is:

$$\widehat{Pr}(P \leq \gamma) = \frac{\#\{p_i \leq \gamma\}}{G} = \frac{R(\gamma)}{G},$$

where $R(\gamma) = \#\{p_i \leq \gamma\}$. When $R(\gamma) = 0$, the estimate would be undefined, therefore, $R(\gamma)$ is replaced with $R(\gamma) \vee 1$. Also $1 - (1-\gamma)^G$ is a lower bound for $\Pr(R(\gamma) > 0)$. Therefore, pFDR is estimated as:

$$\widehat{\text{pFDR}}_\lambda(\gamma) = \frac{\hat{\pi}_0(\lambda)\gamma}{\widehat{\Pr}(P \leq \gamma)\{1 - (1-\gamma)^G\}} = \frac{W(\lambda)\gamma}{(1-\lambda)\{R(\gamma) \vee 1\}\{1 - (1-\gamma)^G\}}.$$

Similarly, by this method, FDR can be estimated as:

$$\widehat{\text{FDR}}_\lambda(\gamma) = \frac{\hat{\pi}_0(\lambda)\gamma}{\widehat{\Pr}(P \leq \gamma)} = \frac{W(\lambda)\gamma}{(1-\lambda)\{R(\gamma) \vee 1\}}.$$

They also discussed a pFDR analogue of the p-value, called q-value. Recall for a nested set of rejection regions $\{\Gamma\}$, the p-value of an observed statistic $T = t$ is defined to be

$$\text{p-value}(t) = \min_{\Gamma : t \in \Gamma} \{\Pr(T \in \Gamma | H = 0)\}.$$

For an observed statistic $T = t$, the q-value of t is defined to be

$$\text{q-value}(t) = \min_{\Gamma: t \in \Gamma} \{\text{pFDR}(\Gamma)\}.$$

For a set of hypothesis tests conducted with independent p-values, the q-value of the observed p-value p is:

$$q(p) = \inf_{\gamma \geq p} \{\text{pFDR}(\gamma)\}. \tag{3.5}$$

For the G hypothesis tests, calculate the p-values $p_1, \cdots, p_G$, let $p_{(1)} \leq \cdots \leq p_{(G)}$ be the ordered p-values.

$$\hat{q}(p_{(G)}) = \widehat{\text{pFDR}}(p_{(G)}),$$

$$\hat{q}(p_{(i)}) = \min\{\widehat{\text{pFDR}}(p_{(i)}), \hat{q}(p_{(i+1)})\},$$

for $i = G-1, G-2, \cdots, 1$.

Storey and Tibshirani (2001) [65] estimated pFDR under dependence. Storey and Tibshirani (2003) [66] applied pFDR and q-value to genomewide studies. Theoretical aspects of pFDR have also been developed in these references.

3.2.4 Benjamini-Hochberg and Storey Methods

Benjamini-Hochberg way is to fix the acceptable rate α beforehand and estimate a significance threshold to obtain this rate conservatively on average. Storey way is to fix the significance threshold and provide a conservative estimate of the rate over that threshold. Storey, Taylor and Siegmund (2004) [64] showed that in both finite sample and asymptotic settings that the goals of the two approaches are essentially equivalent. We prefer Benjamini-Hochberg way as it is simple and easy to implement.

3.3 Class Prediction

Classification and prediction are important topics in microarray data analysis. For example, in studies of cancer, it is hoped that gene expression profiling enables us to diagnose tumors precisely and systematically. By studying and contrasting gene expression profiles of normal and tumorous tissues, we can not only gather valuable information regarding the gene

expression pattern of the underlying disease process, but also predict the class of a new tumor based on its gene expression profile. By studying the contribution of each gene to the classification model, we will find a set of good biomarkers.

Depending on the experiment, the N samples may correspond to N tissues, cell lines, tumors, or something else. The N samples belong to k different classes. The class each sample belongs to is known and there are n_i samples from the ith class, $\sum_{i=1}^{k} n_i = N$. The N-vector, $\mathbf{y} = \{y_{ij}\}$, indicates to which class each sample belongs: $y_{ij} = i$ is the classification indicator of the jth sample in the ith class. Let $\mathbf{x}_j^i = (x_{ij1}, \cdots, x_{ijG})$, a G-dimensional column vector, denote the gene expression levels on the jth sample in the ith class. $\overline{\mathbf{x}}_i = \sum_{j=1}^{n_i} \mathbf{x}_j^i / n_i$; $\overline{\mathbf{x}} = \sum_{i=1}^{k} n_i \overline{\mathbf{x}}_i / N$. Generally, one classification rule will partition the space of all possible $\mathbf{x}$'s into k disjoint subsets, $A_1, \cdots, A_k$, such that if $\mathbf{x}$ falls into A_m, then $\mathbf{x}$ is predicted to belong to class m.

3.3.1 Linear Discriminant Analysis

Linear discriminant analysis (LDA) is one of the oldest and simplest methods of supervised classification. It calculates the projection matrix $W_{G\times d}$ that maximizes the Fisher's Linear Discriminant criterion:

$$W_{opt} = \arg\max_{W} \frac{|W^T S_b W|}{|W^T S_w W|}, \tag{3.6}$$

where

$$S_b = \frac{1}{N} \sum_{i=1}^{k} n_i (\overline{\mathbf{x}}_i - \overline{\mathbf{x}})(\overline{\mathbf{x}}_i - \overline{\mathbf{x}})^T, \tag{3.7}$$

is the between-class scatter matrix and

$$S_w = \frac{1}{N} \sum_{i=1}^{k} \sum_{j=1}^{n_i} (\mathbf{x}_j^i - \overline{\mathbf{x}}_i)(\mathbf{x}_j^i - \overline{\mathbf{x}}_i)^T = \frac{1}{N} \sum_{i=1}^{k} (n_i - 1) S_i, \tag{3.8}$$

is the within-class scatter matrix. It aims to simultaneously minimizes the within-class distance and maximize the between-class distances in order to achieve maximum discrimination. If S_w is a non-singular matrix, $S_w^{-1} S_b$ has at most $d = \min(k-1, G)$ non-zero eigenvalues. W_{opt} is consisted of the corresponding eigenvectors $\mathbf{w}_1, \cdots, \mathbf{w}_d$. For a new sample $\mathbf{x}$, the predicted class is

$$C(\mathbf{x}) = \arg\min_{k} \sum_{i=1}^{d} (\mathbf{w}_i^T (\mathbf{x} - \overline{\mathbf{x}}_k))^2.$$

Although LDA performs well in many applications, we cannot directly use LDA for microarray data analysis because of the small sample size problem, for which S_w is singular.

Tian et al. (1986) [69] tried to replace the inverse S_w^{-1} with the pseudo-inverse S_w^+:

$$\begin{aligned} M &= ((\mathbf{x}_1^1 - \overline{\mathbf{x}}_1), (\mathbf{x}_2^1 - \overline{\mathbf{x}}_1), \cdots, (\mathbf{x}_{n_1}^1 - \overline{\mathbf{x}}_1), \\ &\quad (\mathbf{x}_1^2 - \overline{\mathbf{x}}_2), (\mathbf{x}_2^2 - \overline{\mathbf{x}}_2), \cdots, (\mathbf{x}_{n_2}^2 - \overline{\mathbf{x}}_2), \\ &\quad \cdots\cdots\cdots\cdots \\ &\quad (\mathbf{x}_1^k - \overline{\mathbf{x}}_k), (\mathbf{x}_2^k - \overline{\mathbf{x}}_k), \cdots, (\mathbf{x}_{n_k}^k - \overline{\mathbf{x}}_k)) \\ S_w^+ &= M(M^T M)^{-2} M^T \end{aligned} \tag{3.9}$$

To handle the singularity problem, Hong and Yang (1991) [45] added a singular value perturbation to S_w to make it nonsingular based on the following theorem:

Theorem 1. *Let $A = U\Sigma V^T$ be an $n \times n$ matrix, rank(A)= r. ϕ represents the set of $n \times n$ matrices whose rank is less or equal to k, $0 < k < r$. If X is a matrix in ϕ and it satisfies the condition:*

$$\|A - X\|_F = \min_{S \in \phi} \|A - S\|_F$$

then

$$\|A - X\|_F = \sqrt{\lambda_{k+1} + \lambda_{k+2} + \cdots + \lambda_n}$$

where $\sqrt{\lambda_i}$ is a singular value of A and X. $\|\cdot\|_F$ is the Frobenius norm which is a unitary invariance norm.

Denote singular value decomposition (SVD) of S_w as: $S_w = U\Sigma V^T$ where $\text{rank}(S_w) = r$, Σ =diagonal$\{\sqrt{\lambda_1}, \sqrt{\lambda_2}, \cdots, \sqrt{\lambda_r}, 0, \cdots, 0\}$ and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r$; U and V are orthonormal matrices. λ_i is the nonzero eigenvalue of matrix $(S_w S_w^T)$; it is also the eigenvalue of $(S_w^T S_w)$. $\sqrt{\lambda_i}$ is called the singular value of S_w. Any column vector u_i and v_i of U, V are the eigenvectors which correspond to the eigenvalue λ_i of $(S_w S_w^T)$ and $(S_w^T S_w)$ respectively. A small perturbation Δ is added to S_w: $S_w + \Delta = \overline{S}_w$, such that

$$\overline{S}_w = U\overline{\Sigma}V^T,$$

where $\overline{\Sigma}$ =diagonal$\{\sqrt{\lambda_1}, \sqrt{\lambda_2}, \cdots, \sqrt{\lambda_r}, \sqrt{\delta}, \cdots, \sqrt{\delta}\}$, and $0 < \delta \leq \lambda_r$. Then according to the above theorem, with Frobenius norm, $\overline{S}_w$ is a nonsingular matrix which is the closest to S_w and the difference between S_w and $\overline{S}_w$ is $\sqrt{(n-r)\delta}$.

Friedman (1989) [30] expressed LDA by assuming that samples from class i follow some distribution with density function $f_i(\mathbf{x})$:

$$f_i(\mathbf{x}) = (2\pi)^{-G/2}|\Sigma_i|^{-1/2}\exp[-\frac{1}{2}(\mathbf{x}-\mu_i)^T\Sigma_i^{-1}(\mathbf{x}-\mu_i)], \tag{3.10}$$

where μ_i and Σ_i are the class $i(1 \le i \le k)$ population mean vector and covariance matrix. The classification rule is: choose $\hat{i}$ such that

$$\hat{i} = \arg\max_{1\le i\le k} f_i(\mathbf{x})\pi_i$$

where π_i is the unconditional prior probability of observing a class i member, which is equivalent to

$$\hat{i} = \arg\min_{1\le i\le k}(\mathbf{x}-\mu_i)^T\Sigma_i^{-1}(\mathbf{x}-\mu_i) + \ln|\Sigma_i| - 2\ln\pi_i.$$

When all of the class covariance matrices are presumed to be identical:

$$\Sigma_i = \Sigma,\ 1 \le i \le k, \tag{3.11}$$

it is linear discriminant analysis (LDA). Without the assumption (3.11), the above classification rule is called quadratic discriminant analysis (QDA). To obtain more reliable estimates of the eigenvalues in the sample covariance matrix, a ridge-type regularized estimate $\hat{\Sigma}_i(\lambda,\gamma)$ was proposed:

$$\hat{\Sigma}_i(\lambda,\gamma) = (1-\gamma)\hat{\Sigma}_i(\lambda) + \frac{\gamma}{G}tr[\hat{\Sigma}_i(\lambda)]I,$$

where

$$\hat{\Sigma}_i(\lambda) = \frac{(1-\lambda)S_i + \lambda\sum_{i=1}^k S_i}{(1-\lambda)n_i + \lambda N}.$$

The classification rule with $\hat{\Sigma}_i(\lambda,\gamma)$ is called regularized discriminant analysis (RDA), which is also a compromise between LDA and QDA.

Hastie et al. (1995) [40] proposed to replace Σ_w by a regularized version $\Sigma_w + \lambda\Omega$, where Ω is a penalty matrix. They called this method penalized discriminant analysis (PDA), whose purpose is to impose a spatial smoothness constraint on the coefficients, which is not necessary for microarray analysis.

Dudoit et al. (2002) [26] considered Diagonal Linear Discriminant Analysis (DLDA), where the class densities (3.10) have the same diagonal covariance matrix:

$$\Sigma_i = \mathrm{diag}(\sigma_1^2, \cdots, \sigma_G^2)$$

for all $1 \leq i \leq k$. Then the classification rule is:

$$C(\mathbf{x}) = \arg\min_{1 \leq j \leq k} \frac{\|\mathbf{x} - \overline{\mathbf{x}}_j\|^2}{\sigma_j^2}.$$

They pointed out that the "weighted gene voting scheme" for binary classification in Golub et al. (1999) [37] was a variant of DLDA using maximum likelihood estimation.

Li et al. (2003) [48] proposed a new feature extraction criterion, the maximum margin criterion (MMC), to avoid the small sample size problem. From a geometric standpoint, MMC maximizes the (average) margin between classes.

$$W_{opt} = \arg\max_{W} tr(W^T(S_b - S_w)W).$$

Li et al. (2005) [49] developed a generalized linear discriminant analysis (GLDA) that is a general, direct, and complete solution to optimize Fisher's criterion. They wrote the Fisher's criterion as:

$$W_{opt} = \arg\max_{W} tr\left(\frac{|W^T S_b W|}{|W^T S_t W|}\right), \tag{3.12}$$

where

$$S_t = S_b + S_w. \tag{3.13}$$

This equation is equivalent to (3.6) when S_w is nonsingular. They used Moore-Penrose inverse of singular S_w.

Definition 1 (Moore-Penrose Inverse)**.** *A matrix A^+ satisfying the following conditions is unique and is called the Moore-Penrose inverse of A:*

$$\begin{aligned} AA^+A &= A \\ A^+AA^+ &= A^+ \\ (A^+A)^T &= A^+A \\ (AA^+)^T &= AA^+. \end{aligned}$$

They showed that the Fisher's criterion (3.12) is maximized by the largest eigenvectors of $S_t^+ S_b$. Their algorithm is summarized as follows:

1. Calculate $M = [\sqrt{p_1}(\overline{\mathbf{x}}_1 - \overline{\mathbf{x}}), \cdots, \sqrt{p_k}(\overline{\mathbf{x}}_k - \overline{\mathbf{x}})]$ and $X = \frac{1}{\sqrt{N}}[(\mathbf{x}_1 - \overline{\mathbf{x}}), \cdots, (\mathbf{x}_N - \overline{\mathbf{x}})]$;
2. Perform the SVD $X = U\Lambda^{1/2}V^T$;
3. $S_t^{+\frac{1}{2}} = U\Lambda^{-\frac{1}{2}}U^T$;
4. Perform the SVD $S_t^{+\frac{1}{2}}M = \widetilde{U}\widetilde{\Lambda}^{\frac{1}{2}}\widetilde{V}^T$;
5. $W = S_t^{+\frac{1}{2}}\widetilde{U}$;

where SVD represents singular value decomposition.

Sharma and Paliwal (2008) [59] proposed to optimize the Fisher's criterion by using gradient descent algorithm, which they called Gradient LDA (GLDA). The GLDA algorithm is as follows:

1. $$W \leftarrow W - \alpha\frac{\partial\hat{J}(W)}{\partial W};$$
2. $W \leftarrow$ Normalize each of the column vectors of W separately;

where $\hat{J}(W) = 1/J(W)$,

$$\frac{\partial\hat{J}(W)}{\partial W} = 2\hat{J}(W)[S_wW(W^TS_wW)^{-1} - S_bW(W^TS_bW)^{-1}]$$

and $\alpha > 0$ is the learning rate parameter.

3.3.2 Reducing the High Dimensionality

Classical classification techniques assume that the number of samples are larger than the number of features (variables, genes here) in the data. But most of microarray data consist of a much larger number of features (genes) compared to the number of samples. Thus, to apply classical classification methods directly, like LDA, we need to reduce the dimension first.

As Sharma and Paliwal (2008) [59] pointed out, different methods used for dimension reduction can be grouped into two categories: feature extraction methods and feature selection methods. While feature extraction methods construct a few features from the large number of original features through their linear or nonlinear combination, feature selection (or feature filtering) methods retain original useful features and discard others.

Feature Extraction

Principal component analysis (PCA) and partial least squares (PLS) are often applied to reduce the high dimensionality in microarray data. Denote the microarray data as a matrix:

$$X = (\mathbf{x}_1^1, \mathbf{x}_2^1, \cdots, \mathbf{x}_{n_1}^1, \mathbf{x}_1^2, \mathbf{x}_2^2, \cdots, \mathbf{x}_{n_2}^2, \cdots\cdots, \mathbf{x}_1^k, \mathbf{x}_2^k, \cdots, \mathbf{x}_{n_k}^k).$$

The correlation matrix is:

$$R_{G\times G} = \frac{1}{N-1} X(I_{G\times G} - \frac{1}{G}\mathbf{1}_{G\times 1}\mathbf{1}_{G\times 1}^T)X^T.$$

Denote the eigenvalue decomposition of R as $R = U\Lambda U^T$, where Λ =diag$\{\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{N-1}\}$ are the ordered nonzero eigenvalues, and $U = \{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_{N-1}\}$ are the corresponding eigenvectors. The ith principal components is a linear combination of the original variables, $X^T\mathbf{u}_i$.

The objective equation of PCA could be written as follows:

$$\mathbf{u}_s = \underset{\mathbf{u}^T\mathbf{u}=1}{\arg\max}\, var(X^T\mathbf{u}) \tag{3.14}$$

subject to the orthogonality constraint

$$cov(X^T\mathbf{u}, X^T\mathbf{u}_j) = 0, \text{ for all } 1 \leq j < s.$$

The maximum number of nonzero components is the rank of X. As the objective equation (3.14) does not involve the response variable, it does not necessarily yield components predictive of the response variable. To include the response variable in the objective equation, PLS was introduced. The objective equation of PLS is:

$$\mathbf{w}_s = \underset{\mathbf{w}^T\mathbf{w}}{\arg\max}\, cov^2(X^T\mathbf{w}, \mathbf{y}) \tag{3.15}$$

subject to the orthogonality constraint

$$cov(X^T\mathbf{w}, X^T\mathbf{w}_j) = 0, \text{ for all } 1 \leq j < s.$$

The maximum number of components is also the rank of X. The ith PLS components are also a linear combinations of the original predictors, $X\mathbf{w}_i$. As Nguyen and Rocke (2002) [52] pointed out, it is reasonable to suspect that if the original variables (genes) are already

predictive of response classes, then the constructed components from PCA would likely be good predictors of response classes. But they did report an improvement in classification by using PLS rather than PCA for microarray data.

Belhumeur et al. (1997) [8] and Khan et al. (2001) [47] applied principal component analysis (PCA) for dimensionality reduction. This method is criticized to be sub-optimal that PCA step may discard dimensions that contain important discriminative information.

Chen et al. (2000) [25] pointed out that the null space of S_w (3.8) carries most of the discriminative information: for a projection direction a, if $S_w a = 0$, and $S_b a \neq 0$, then

$$\frac{a^T S_b a}{a^T S_w a}$$

is maximized. They proposed a two-stage procedure:

1. Suppose $V_0 = span\{\alpha_i | S_w \alpha_i = 0,\ \alpha_i \in R^G,\ |\alpha_i| = 1,\ \alpha_i^T \alpha_j = 0,\ i \neq j,\ i = 1, \cdots, G - r\}$ where $r = rank(S_w)$, i.e. V_0 is the null space of S_w. Denote $Q = \{\alpha_1, \alpha_2, \cdots, \alpha_{G-r}\}$. Projecting the data into V_0, we get the within-class and between-class scatter matrix of the transformed samples in V_0 ($\widetilde{S}_b$, $\widetilde{S}_w$):

$$\widetilde{S}_b = QQ^T S_b QQ^T,$$

$$\widetilde{S}_w = QQ^T S_w QQ^T = 0.$$

2. Calculate the eigenvectors corresponding to the set of the largest eigenvalues of $\widetilde{S}_b$ and use them to form the most discriminant vector set for LDA.

The drawback of this method is that it needs to compute the rank of S_w, which is an ill-defined operation due to floating-point imprecision, and that the computational complexity of determining the null space of S_w is very high.

Yu and Yang (2001) [78] provided a direct LDA algorithm for high-dimensional data. The key idea of their algorithm is to discard the null space of S_b (3.7) - which contains no useful information - rather than discarding the null space of S_w, which contains the most discriminative information. This can be achieved by diagonalizing S_b first and then diagonalizing S_w. The traditional procedure takes the reverse order. Both approaches produce the same result when S_w is not singular. The algorithm is outlined below:

1. Diagonalize S_b: find matrix V such that $V^T S_b V = \Lambda$, where $V^T V = I$ and Λ is a diagonal matrix sorted in decreasing order. Let Y be the first m columns of V, $Y^T S_b Y = D_b > 0$, where D_b is the $m \times m$ principal sub-matrix of Λ.

2. Let $Z = Y D_b^{-\frac{1}{2}}$, $Z^T S_b Z = I$. Diagonalize $Z^T S_w Z$ by eigen-analysis: $U^T Z^T S_w Z U = D_w$, where $U^T U = I$.

3. Let the LDA matrix $A = U^T Z^T$. $A S_w A^T = D_w$, $A S_b A^T = I$.

4. The final transformation that spheres the data should be

$$x^\star \leftarrow D_w^{-\frac{1}{2}} A x.$$

In Direct LDA, one may also employ S_t instead of S_w. In this way, Direct LDA is actually equivalent to the PCA+LDA. Therefore, Direct LDA may be regarded as a "unified PCA-LDA" since there is no separate PCA step.

Huang et al. (2002) [46] first removed the null space of S_t (3.13), which is the common null space of S_b and S_w and has no use in discrimination analysis, and then LDA-PCA was performed in the lower-dimensional projected space. Their procedures are as follows:

1. Remove the null space of S_t: do eigen-analysis on the $N \times N$ matrix $\frac{1}{N} M_t^T M_t$ where

$$\begin{aligned} M_t \;=\; & ((\mathbf{x}_1^1 - \overline{\mathbf{x}}), (\mathbf{x}_2^1 - \overline{\mathbf{x}}), \cdots, (\mathbf{x}_{n_1}^1 - \overline{\mathbf{x}}), \\ & (\mathbf{x}_1^2 - \overline{\mathbf{x}}), (\mathbf{x}_2^2 - \overline{\mathbf{x}}), \cdots, (\mathbf{x}_{n_2}^2 - \overline{\mathbf{x}}), \\ & \cdots\cdots\cdots\cdots \\ & (\mathbf{x}_1^k - \overline{\mathbf{x}}), (\mathbf{x}_2^k - \overline{\mathbf{x}}), \cdots, (\mathbf{x}_{n_k}^k - \overline{\mathbf{x}})) \end{aligned} \tag{3.16}$$

   and $S_t = M_t M_t^T$; let U be the matrix whose columns are all the eigenvectors of S_t corresponding to the nonzero eigenvalues, then: $S'_w = U^T S_w U$ and $S'_b = U^T S_b U$.

2. Calculate the null space of S'_w: Q, then $S''_w = Q^T S'_w Q = 0$ and $S''_b = Q^T S'_b Q = (UQ)^T S_b (UQ)$.

3. Remove the null space of S''_b if it exists: let V be the matrix whose columns are all the eigenvectors of S''_b corresponding to the nonzero eigenvalues or part of them associated with the largest eigenvalues, then the final LDA projection is: $W = UQV$.

Lu et al. (2005) [50] merged LDA and PCA in a unified framework, which they called hybrid PCA and LDA analysis, with the following objective equation:

$$W_{opt} = \arg\max_{W} \frac{|W^T[(1-\lambda)\cdot S_b + \lambda \cdot S_t]W|}{|W^T[(1-\eta)\cdot S_w + \eta \cdot I]W|}, \tag{3.17}$$

where the range of the parametric pair (λ, η) is from $(0,0)$ to $(1,1)$. When $(\lambda = 0, \eta = 0)$, (3.17) reduces to LDA; when $(\lambda = 1, \eta = 1)$, (3.17) reduces to PCA; when $(\lambda = 0, \eta = 1)$, (3.17) maximizes the scatters among the classes with minimal effort on clustering each class; when $(\lambda = 1, \eta = 0)$, (3.17) minimizes the scatter matrices of within-classes.

Feature Selection

Feature selection (filtering) is done by ranking all the genes according to some statistics. When the statistics used in selecting feature do not involve the response variable, for example, the overall variance and overall mean across all samples (Amaratunga and Cabrera (2003) [1] and Talloen et al. (2007) [68]), the methods are called "unsupervised feature selection". When they do involve the response variable, for example, an ordinary t statistic, or statistics "borrowing strength" amongst genes such as conditional t (Amaratunga and Cabrera (2009) [2]), or LIMMA t (Smyth (2004) [61]), or SAM t statistic (Tusher et al. (2001) [71]), or score statistic (Bair et al. (2006) [6]), the methods are called "supervised feature selection".

Golub et al. (1999) [37] did gene selection through their degree of correlation with the class distinctions. They measured the correlation by the 'signal-to-noise' ratio. Denoting the means and standard deviations of the log of the expression level of gene g as $(\mu_1(g), \sigma_1(g))$ and $(\mu_2(g), \sigma_2(g))$ respectively, they define the 'signal-to-noise' ratio as

$$P(g) = [\mu_1(g) - \mu_2(g)]/[\sigma_1(g) - \sigma_2(g)].$$

Large values of $|P(g)|$ indicate a strong correlation. The sign of $P(g)$ being positive or negative corresponds to g being more highly expressed in class 1 or class 2. They ranked the genes by the absolute values of the correlation coefficients, selected equal number of genes with positive and negative correlation, and did classification by 'weighted voting', where each informative gene casts a vote for one of the classes and the magnitude of each

vote is decided based on the degree of correlation. Slonim et al. (2000) [60] applied the same selection procedure. Furey et al. (2000) [34] applied Golub's selection method, but selected genes only by the absolute values of the correlation coefficients.

Guyon et al. (2002) [38] also preferred gene selection methods. They claimed that it is of practical importance to select a small subset of genes to build diagnostic tests. It is also cost-efficient and will be easy to verify the relevance of selected genes. But they found that these feature ranking with correlation methods are based some implicit orthogonality assumptions, since each coefficient is computed without taking into account mutual information between genes. Instead, they brought out one feature subset ranking criterion using Recursive Feature Elimination (RFE). They treated each subset like a model and ranked each subset using some model selection criterion. Dudoit et al. (2002) [26] selected genes by somewhat random procedures: bagging and boosting.

Cautions about feature selection methods are: (1) selecting too many genes may not reduce the dimensionality enough, while selecting too few genes may lose too much information; (2) feature selection based on methods studying each gene separately may ignore the fact that genes are related. Bo and Jonassen (2002) [16] showed that a pair of genes in combination separates two classes better than doing the filtering gene by gene. Gene pairs (or other multiples) can be selected using Hotelling's t test.

As Amaratunga and Cabrera (2003) [1] and Becker et al. (2011) [7] pointed out that a pre-set limit on the number of features to be chosen may exclude some informative genes and result in bad performance, feature filtering is generally implemented in this way: genes whose statistics are larger than some threshold will be picked up. How to determine the threshold remains one issue for filtering methods. Cross validation could be applied, but it is not appealing in genetic association studies since most microarray data have small sample sizes.

Penalization

In reviewing penalized method here, to simplify the notations, we assume that $k = 2$, i.e., the samples are grouped into 2 classes. To avoid overfitting in high dimensional settings, different penalties were introduced into the objective functions to shrink the coefficients

of the variables (genes) toward 0, hence achieve the reduction of dimensionaltiy or feature selection. Here are some popular penalties:

LASSO: Lasso penalty was introduced in Tibshirani (1996) [70]. By imposing an L_1 penalty on the regression coefficients, the lasso does both continuous shrinkage and automatic variable selection simultaneously. The penalty function is as follows:

$$P^L(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1; \tag{3.18}$$

Elastic-Net: Zou and Hastie (2005) [80] pointed out that the lasso selects at most N variables when $G > N$, which makes L_1 penalty inappropriate for microarray data. Also if a group of variables are highly correlated, the lasso tends to select one from the group and ignore the others. To address this problem, they proposed the elastic net penalty function, which is a combination of quadratic penalty and the L_1 penalty:

$$P_\alpha^{EN}(\boldsymbol{\beta}) = (1-\alpha)\cdot\frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + \alpha\cdot\|\boldsymbol{\beta}\|_1; \tag{3.19}$$

SCAD: On the other hand, the lasso is criticized for the significant bias toward 0 for large regression coefficients. To diminish the bias, Fan and Li (2001) [28] proposed the smoothly clipped absolute deviation (SCAD) penalty:

$$p_{a,\lambda}^{SCAD}(\beta) = \begin{cases} \lambda|\beta| & \text{if } |\beta| \le \lambda, \\ -\frac{|\beta|^2 - 2a\lambda|\beta| + \lambda^2}{2(a-1)} & \text{if } \lambda < |\beta| \le a\lambda \\ \frac{(a+1)\lambda^2}{2} & \text{if } |\beta| > a\lambda \end{cases}, \tag{3.20}$$

for $\lambda \ge 0$ and $a > 2$. It is symmetric, nonconvex, and singular at the origin. The SCAD penalty has the same form as the lasso penalty at the neighborhood of 0, and applies a constant penalty for large coefficients.

MCP: Zhang (2010) [79] proposed the minimax concave penalty (MCP):

$$p_{a,\lambda}^{MCP}(\beta) = \begin{cases} \lambda|\beta| - \frac{|\beta|^2}{2a} & \text{if } |\beta| \le a\lambda, \\ \frac{a\lambda^2}{2} & \text{if } |\beta| > a\lambda \end{cases}, \tag{3.21}$$

with $\lambda \ge 0$ and $a > 1$. It is symmetric, nonconvex, and singular at the origin. The MCP relaxes the penalization continuously until $|\beta| \le a\lambda$, then it applies a constant penalty.

Regression models with SCAD or MCP have oracle property: the performance is asymptotically as good as if one knows the true positions of nonzero coefficients.

The linear logistic regression model is one popular method for classification and prediction:

$$p(\mathbf{x}) = \Pr(y = 1|\mathbf{x}) = \frac{1}{1 + \exp(-(\beta_0 + \mathbf{x}\boldsymbol{\beta}))},$$

where $\beta_0, \boldsymbol{\beta}$ are the coefficients. The penalized objective function is as follows:

$$\max_{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}^{G+1}} \left[\frac{1}{N} \sum_{i=1}^{N} \left(I_{(y_i=1)} \log p(\mathbf{x}_i) + I_{(y_i=-1)} \log(1 - p(\mathbf{x}_i)) \right) - \lambda P(\boldsymbol{\beta}), \right] \tag{3.22}$$

where $P(\boldsymbol{\beta})$ could be the lasso (3.18), elastic-net (3.19), SCAD (3.20) or MCP (3.21) penalty. For the lasso and elastic-net penalties, Zou and Hastie (2005) [80] tried to find the optimum by steepest descent which can be time consuming. Friedman et al. (2010) [31] proposed the Glmnet algorithm, which is quite efficient for solving this problem. Breheny and Huang (2011) [19] applied coordinate descent algorithm to solve the nonconvex regression problems with MCP or SCAD penalty along with a L_2 penalty.

Support vector machine (SVM) is one of the most powerful supervised classification techniques, which separates two classes by a hyperplane with maximum margin. For non-separable data, the soft-margin SVM uses slack variables to control the upper bound of the misclassification error. Since $G \gg N$, our data can always be separated by a hyperplane. Also Hastie et al. (2001) [42] pointed out: linear classifiers often give better performances than nonlinear ones in many applications with $G \gg N$, even though nonlinear classifiers are known to be more flexible. Hence, we only consider linear SVM here, which finds a hyperplane $f(\mathbf{x}) = b + \mathbf{w} \cdot \mathbf{x}$ by minimizing

$$\frac{1}{N} \sum_{i=1}^{N} [1 - y_i(b + \mathbf{w} \cdot \mathbf{x}_i)]_+ + \lambda \|\mathbf{w}\|^2, \tag{3.23}$$

where b is a constant, $\mathbf{w}$ is the directional vector, and $\mathbf{w}\cdot\mathbf{x} = \sum_{j=1}^{G} w_j x_j$. The quantity $\|\mathbf{w}\|^2$ is the inverse of the squared width of the margin for the classifier. The tuning parameter λ controls the trade-off between minimizing the loss function and maximizing the margin. The hinge loss $[1 - y_i(b + \mathbf{w} \cdot \mathbf{x}_i)]_+$ is a convex upper bound for the 0-1 loss function.

Even with linear SVM, a relatively large sample size is needed, otherwise the spurious signal will generate a huge margin that is almost unbeatable by the good signal. Suppose

that we have a typical microarray experiment with 5+5 observations and 45000 genes. By a single gene, there are $C_{10}^5/2 = 126$ ways that the 10 samples are separated into two groups with equal sample size. Then by random order, the probability that one gene separates the two groups correctly is 1/126. Since we have 45000 genes, on average there would be $45000/126 \doteq 357$ genes that will separate perfectly the two groups just by chance. Each of these genes has a corresponding separation margin, and the best will be the biggest. In addition the combination of these genes would form a huge margin, which could be much larger than the one produced by the good signal. To improve the performance of SVM, different methods were proposed.

Fung and Mangasarian (2001) [32] brought out a new formulation:

$$\min_{b,\mathbf{w}} \lambda \sum_{i=1}^{N} [1 - y_i(b + \mathbf{w} \cdot \mathbf{x}_i)]^2 + \mathbf{w}'\mathbf{w} + b^2, \tag{3.24}$$

which they call "linear proximal SVM". The advantage of this formulation is that it has an explicit exact solution and can be solved by 6 lines of MATLAB code. As pointed out in Hastie et al. (2001) [42], the objective function (3.23) has the "loss and penalty" form. From this perspective, the various penalties used in regression can be applied in SVM. The L_1-SVM was proposed by Bradley and Mangasarian (1998) [18]. Fung and Mangasarian (2004) [33] developed a fast Newton algorithm to solve the dual problem of the L_1 SVM. Wang et al. (2006) [72] adapted the Elastic Net penalty to SVM (Elastic Net SVM), whose optimization problem is:

$$\min_{b,\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} [1 - y_i f(\mathbf{x}_i)]_+ + \lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \|\mathbf{w}\|_2^2,$$

whereas $\lambda_1, \lambda_2 \geq 0$ are the corresponding tuning parameters. Becker et al. (2011) [7] showed that Elastic SCAD SVM using the following penalty

$$pen_\lambda(\mathbf{w}) := \sum_{i=1}^{G} p_{a,\lambda_1}^{SCAD}(w_i) + \lambda_2 \|\mathbf{w}\|_2^2, \tag{3.25}$$

provided sparser classifiers in terms of median number of features selected and better predictions in terms of misclassification error than Elastic Net SVM.

Penalized methods are used to shrink the coefficients of some genes by the penalties in the objective functions. This implements variable selection automatically and could

improve the performance of prediction. There are algorithms for efficiently computing the penalized methods. However, since penalized methods do variable selection based on the prediction performance, they could select a group of non-specific genes that predict well in the existing data but poor for future data. For example, suppose in one unbalanced experiment, the control group consists of all men, while the treatment group consists of all women. Therefore, the genes related to gender will give a perfect prediction, and could easily be selected by penalized methods. But the performance of the model relying heavily on gender genes would be terrible for future data. In addition, the procedure of picking up genes by penalized methods is a black box and what scientists and statistician can do after fixing the model is quite limited.

Chapter 4

Enriched PCA-LDA

4.1 Introduction

In microarray experiments, the expression levels of tens of thousands of genes are monitored simultaneously. But because of the high cost, the sample size is typically small: only a few dozen. In most cases, only a relatively small proportion out of the tens of thousands of genes carry information related to the response. As we have reviewed in chapter 2, there are different methods for classification of microarray data. In summary, the methods could be classified into two categories: one is two-step method, which reduces the high dimensionality by feature selection or feature extraction and does the classification by classical methods such as LDA and SVM; the other is penalized method which integrates the dimension reduction step and the classification step together. Here, we introduce a new perspective on microarray data analysis: though microarray data have quite complicated structures, we believe they generally contain three types of signals - specific signal, non-specific signal and spurious signal.

Specific signal is the signal truly associated with the response being studied and is generated by the process that creates this response. For example, in a cancer screening study, specific signal is the one generated by the genes involved in the cancer process. Non-specific signal is due to some secondary effect unrelated to the experiment in the population but it is possible that in a particular dataset such an effect would be correlated with the response. For example, in data from unbalanced designs, covariates such as gender, blood type, race, age, and disease severity, which have nothing to do with the response, may introduce some signal because the unbalancedness induces a correlation of the covariate with the response. These two types of signals are hard to separate at the data analysis stage but could be separated with new data that does not have the same unbalanced design

or via subject-matter knowledge. For example, a group of apparently significant genes may turn out to be from the X/Y chromosomes indicating that the signal is related to gender and hence non-specific. Spurious signal is due to the large number of genes in relation to the sample size. The huge number of noise components could generate a random pattern, which appears to be a signal.

Spurious signal in microarray data is typically very strong. It may conceal the other two types of signal. Take one simulation in Bair et al. (2006) [6] as an example. Let x_{gj} denote the "expression level" of the gth gene in the jth sample. The data (which contains 5000 genes and 100 samples) are generated according to the following model:

$$x_{gj} = \begin{cases} 3 + \epsilon_{gj} & \text{if } g \leq 50, j \leq 50 \\ 4 + \epsilon_{gj} & \text{if } g \leq 50, j > 50 \\ 3.5 + 1.5 \cdot I(u_{1j} < 0.4) + \epsilon_{gj} & \text{if } 51 \leq g \leq 100 \\ 3.5 + 0.5 \cdot I(u_{2j} < 0.7) + \epsilon_{gj} & \text{if } 101 \leq g \leq 200 \\ 3.5 - 1.5 \cdot I(u_{3j} < 0.3) + \epsilon_{gj} & \text{if } 201 \leq g \leq 300 \\ 3.5 + \epsilon_{gj} & \text{if } g \geq 301 \end{cases} \tag{4.1}$$

where the random variables u_{gj}'s independently follow uniform$(0, 1)$, the ϵ_{gj}'s are independent standard normal variables, and $I(x)$ is an indicator function. We assume the first 50 samples are in group 1, and the last 50 samples in group 2. Then according to our simulation model (4.1), the signal from the first 50 genes is related to the classification, while the signals from genes 51-100, 101-200, 201-300 are not. In other words, the simulated data contains 1 specific signal, 3 non-specific signals (shown in Figure 4.3) and perhaps some spurious signals. The singular values of the 5 dominant principal components are 97.52, 88.08, 82.04, 79.98,and 79.73 respectively, and the first two components correspond to two non-specific signals. The third principal component with singular value 82.04 is the one related to the classification (see Figure 4.4), which is just above that of the first spurious signal: 79.73. As dimension increases, spurious signals may ultimately dominate some good signals (see Figure 4.6).

Hence, in analysis of microarray data, weakening the spurious signal and highlighting the specific and non-specific signals are appealing. In this chapter, we propose a new method which can achieve this purpose, and can also separate the specific and non-specific signals

when the non-specific signal is not parallel to the specific one. Our method performs a soft-filtering to reach the weakening and strengthening purpose: instead of setting a threshold to do hard-filtering (pick up or discard), we assign general weights (real numbers from 0 to 1; not 0 or 1 weights as in hard-filtering) to all genes. In order to find out the signals, we apply principal component analysis (PCA) to the weighted data. After strengthening the specific signal by weighting, different principle components can be reranked and the specific signal goes up. This helps improving the performance of prediction, which will be done by linear discriminant analysis (LDA) on the space of a few principal components. We call this method "Enriched PCA-LDA".

Just as LDA and logistic regression for low dimensional classification, enriched PCA-LDA can be viewed as the corresponding method of LDA compared to penalized logistic regression in high dimensional classification problems. The penalized logistic regression which is solved by coordinate descent algorithm has more of the flavor of enriched methods. It starts by looking at variables individually and discards those which have no prediction power. Then it does not optimize the objective function following a generic steepest descent direction, instead, it follows only coordinate directions. When the penalty constraint is meet, it stops and goes to the next step with the next constraint value. The main advantage of enriched PCA-LDA comparing to penalized logistic regression is: results from enriched PCA-LDA can be interpreted easily, and sometimes help scientists digging out interesting things.

The rest of this chapter is organized as follows. In section 4.2, we give the detail of our method. We validate our method by simulation in section 4.3. Then, in section 4.4, we analyze Sialin data from a new perspective, compare these methods using this data and also give some results about the data.

4.2 Enriched PCA-LDA for Classification and Prediction

Here are a few notations which will be used later. Given a training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i = (x_{1i}, \cdots, x_{Gi}) \in \mathbb{R}^G$ is the input vector (x_{ki} is the expression level of gene k in the ith sample) and $y_i \in \{+1, -1\}$ indicates the group label, where $y_i = 1$ means sample i is in

group 1 and $y_i = -1$ means sample i is in group 2. Denote the data as a matrix: $X_{G\times n} = (\mathbf{x}_1^T, \mathbf{x}_2^T, \cdots, \mathbf{x}_n^T)$. X is partitioned as $X = (X_1, X_2)$, where X_1 is the gene expression matrix $(G \times n_1)$ of group 1 and X_2 is that of group 2 $(G \times n_2)$ $(n = n_1 + n_2)$.

4.2.1 Our Method: Enriched PCA-LDA

Weighting is the most important part in enriched PCA-LDA. The idea of weighting is similar to "the ABC clustering" (Amaratunga et al. (2008) [4]), "enriched random forest" (Amaratunga et al. (2008) [5]) and "ensemble classifiers" (Amaratunga et al. (2012) [3]). Among all the genes on each microarray, some genes are differentially expressed between groups while some other genes are not. Intuitively, differentially expressed genes help in classification. Hence, we'd like to pay more attention to those differentially expressed genes, i.e., they will receive higher weights. The criterion for differential expression levels is based on hypothesis testing, where the null hypothesis is that the gene is not differentially expressed. We have reviewed different test statistics in section 3.1. When the sample size is small, the ordinary t test is less efficient than the methods that borrow strength amongst genes, such as conditional t test, SAM t test or LIMMA t test.

Here is how we get the weight for each gene. First, we obtain the p values from the hypotheses test. To adjust for multiplicity we will correct the p-values by the False Discovery Rate (FDR) (Benjamini and Hochberg (1995) [9]) as in (3.4), or correct the p-values by the positive False Discovery Rate (pFDR) (Storey (2003) [63]) as in (3.5). The corrected p-values are also called q-values(Storey and Tibshirani (2003) [66]). The q values are effectively adjusted for the spurious signal so the only signal left is the one that is stronger than the signal happening by chance.

Based on the q values, we can define the weight for each gene. What we would want the weights to do is highlighting the genes with relatively small q values and diluting the genes with relatively large q values. Nonnegative decreasing function of q values will do the job. Most often, we use $w_1(q) = -\log(q)$ and $w_2(q) = 1/q - 1$. Since $q \in (0, 1)$ and function $f(x) = 1/x$ explodes at $x = 0$, $w_1(q) = \log(1/q)$ is much more conservative than $w_2(q) = 1/q - 1$ (see Figure 4.1). A small difference for q values at the left tail will be picked up by $w_2(q)$: $w_2(0.01) = 100, w_2(0.009) = 111.11$, while $w_1(0.01) = 4.61$ and

$w_1(0.009) = 4.71$. Further, we know $w_3(q) = \sqrt{1/q} - 1$ is comparable to $w_1(q)$. From $w_3(q)$ we propose a general set of weight functions: $w_\alpha(q) = (1/q)^\alpha - 1, (0 < \alpha < 1)$. We show some of these weight functions in Figure 4.1.

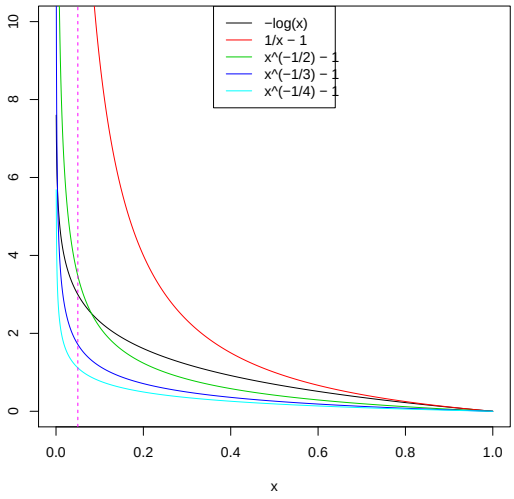

Figure 4.1: Weight functions. The vertical dashed line is cut at $x = 0.05$

The explosion pattern at the neighborhood of 0 of all these functions gives us some freedom to add our preference about when to stop differentiating two q values: say, we may think q values 10^{-7} and 10^{-6} make no difference, we could set a threshold $B_l = 10^{-6}$ and any q values less than B_l will be set as B_l. In this way, situation where only one or two genes dominate will be avoided. In addition, in some conservative circumstances where you prefer to discard no genes, a threshold B_u for q values could be applied: any q values greater than B_u will be set as B_u. Such a B_u is needed in ensemble methods where every gene should have a positive probability to be picked up. Hence, in general, the weight wei'_g for gene g is given by:

$$wei'_g = median(w(B_l), w(q_g), w(B_u)),$$

where B_l and B_u are the lower and upper thresholds for the q values, and $w(q)$ is the weight function you choose. Depending on the nature of the data and the choices of $w(q)$, B_l and B_u, the scale of weights for all the genes could be so big that weighting would introduce some

unnecessary complexity to the data. To avoid such embarrassment, we do standardization:

$$wei_g = \frac{wei'_g}{\sum_{g=1}^{G} wei'_g},$$

where wei_g is the final weight we are going to use for gene g in further analysis.

In Figure 4.2, we show a plot of the q values, and the corresponding weights using $w_1(q)$, $w_2(q)$ and $w_3(q)$. $B_l = 4.5 * 10^{-5}$ for all weight functions. Figure 4.2 shows that $w_1(q) = -\log(q)$ considers the most genes comparing to the other two weight functions. Moreover, by convention, genes with q values less than 0.05 are considered significant. $w_2(q)$ does not give any considerations for q values between 0.01 and 0.05, and even some q values less than 0.01. $w_3(q)$ does consider most of the genes $w_1(q)$ considered. But it gives too much weights to the 3 genes with relatively small q values. Of course, these pictures are for the same B_l. Relatively larger B_l for $w_3(q)$ and even larger B_l for $w_2(q)$ could give weights plots considering the same amount genes as the one for $w_1(q)$. The choice of weight function is still a research question. But it is clear that if one aims at a smaller group of genes, it is better to use $w_\alpha(q), (0 < \alpha \leq 1)$ with small B_l; while if one aims at a larger group, it is better to use the minus log weights, or the $w_\alpha(q), (0 < \alpha \leq 1)$ with large B_l. However, it could be more difficult to justify the choices of B_l for $w_\alpha(q), (0 < \alpha \leq 1)$. Here, we prefer minus log weights for our data, which is more stable: the resulting weights are more spread out.

Once we get the weights, we can do further analysis on the weighted data. Suppose $X_{G\times n}$ ($G \gg n$) is centered so that the columns have zero means. Denote the weighted data as $X^\star = (X_1^\star, X_2^\star) = WX = (WX_1, WX_2)$, where $W = \text{diag}\{wei_1, \cdots, wei_G\}$ is the diagonal matrix of weights.

To reduce the dimensionality, we get the principal components of $X^\star$. Unlike the principal components of X, weighting strengthens the principal component related to y so that the first few (two or three) principal components of $X^\star$ can give sufficient information (verified in section 4.3 by simulation data). Hence, typically we would like to pick the first two or three principal components and no more than $n/2$. Denote the singular value decomposition (SVD) of $X^\star$ as follows:

$$X^\star_{G\times n} = U_{G\times n} D_{n\times n} V^T_{n\times n}. \tag{4.2}$$

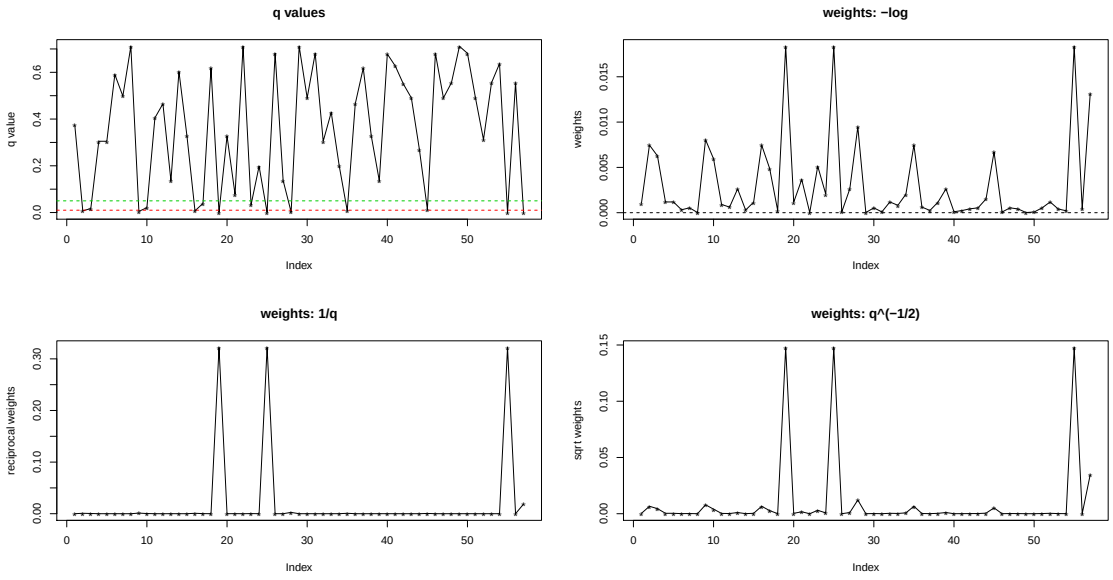

Figure 4.2: The corresponding q values and weights of genes with p values less than 0.001 of our data on day 0. The green dashed line for q values is at 0.05, while the red dashed line is at 0.01.

Projecting $X^\star$ onto the k-dim subspace spanned by the first k columns of U: U_k, we get the first k principal components $Z = U_k^T X^\star$. Then we apply ordinary linear discriminant analysis (LDA) to Z. In order to assess performance, this procedure can be easily combined with cross-validation or splitting the data into one training and one testing sets.

In all, the procedure of our enriched PCA-LDA can be summarized as follows:

1. Get weighted data set $X^\star$ from the weight matrix W and centered X: $X^\star = WX$.
2. Get the projections Z of $X^\star$: the first few eigenvectors of the SVD of $X^\star$.
3. Apply ordinary linear discriminant analysis (LDA) to Z.

The algorithm is very fast, due to the computational efficiency of the SVD of a matrix with a large number of rows but a small number of columns. The computation of the weights is also fast but it comprises the main computational effort of the algorithm.

When a new data vector $\mathbf{x}$ comes in, we first centralize it $\mathbf{x}' = \mathbf{x} - \frac{1}{G}\mathbf{1}_{G\times 1}\mathbf{1}_{G\times 1}^T\mathbf{x}$, then apply the weight matrix W onto the centralized vector $\mathbf{x}'$: $\mathbf{x}'' = W\mathbf{x}'$, then get the projection of $\mathbf{x}''$: $\mathbf{z}'' = U_k^T\mathbf{x}''$, and finally, we predict the class of $\mathbf{z}''$ based on the LDA model of Z, hence, we get the prediction of $\mathbf{x}$.

To weaken the effect of spurious signal, similar to our idea of enriched PCA-LDA, we could do enriched SVM.

4.3 Evaluate our Method based on Simulation

In this section, we evaluate our enriched version of PCA based on simulated data. The simulation model is the same as (4.1). The mean structures of the first four blocks (Each of the first two blocks contain 50 genes, while the third and the forth blocks consist of 100 genes respectively) are shown in Figure 4.3. The first 50 samples are in group 1, and the rest 50 samples are in group 2. Clearly signal similar to the top left plot of Figure 4.3 is what we are pursuing.

The data we get from (4.1) is one 5000×100 matrix. The first to the sixth principal components (PCs) from ordinary principal component analysis (PCA) are plotted in Figure 4.4. We can see that by ordinary principal component analysis the first three principal components show some pattern. But obviously, only the 3rd one will contribute to classification. And the largest four singular values are 97.62, 88.08, 82.04 and 79.98. Hence, the singular value of the most interesting principal component (i.e., the third one) is only slightly larger than the fourth and the fifth ones, which show no pattern and could be some spurious signal.

Our enriched version of PCA was applied to the same data set. The p values were obtained from conditional t test, and were adjusted by FDR as in Benjamini and Hochberg (1995) [9] (3.4). The weight function is $w(q) = -\log(q)$ with $U_l = 10^{-5}$. The corresponding principal components from enriched principal component analysis are in Figure 4.5. We see that weighting in the enriched-PCA strengthened the specific signal: while in ordinary PCA, the PC related to the response is at the third place, and its singular value is only slightly larger than the fourth one, which could belong to some spurious signal; in enriched-PCA, the first PC is the one related to the response (classification), and its singular value is twice as big as the second singular value.

Figure 4.6 and 4.7 are replications of Figure 4.4 and 4.5 with one $45000 * 100$ matrix, which consists of the same $5000 * 100$ matrix as before and another $40000 * 100$ matrix of

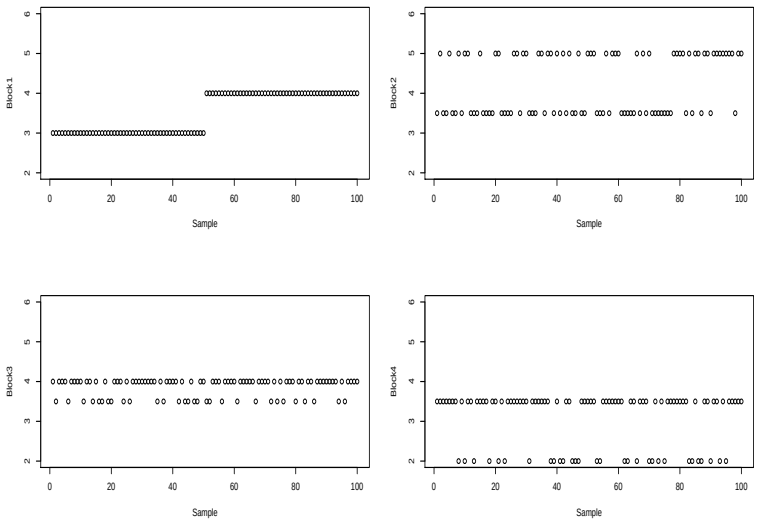

Figure 4.3: The mean of the first four blocks as in model (4.1). As we have designed, the first block is related to the response. The second block has some determinant structure, which does not tell anything about the response. The third and forth blocks have some weak determinant structures.

pure noise. In this case, the ordinary PCA fails to give any meaningful PCs. The signals are concealed by noise. However, our enriched-PCA still works.

4.3.1 Comparing with other Methods

To study how noisy genes could impact ordinary principal component analysis, we tried PCA with data which consists of different number of genes. Data A contains the expression levels of 300 genes: which are those in the first 4 blocks with signals as in model (4.1). Data B contains the expression levels of 500 genes: adding another 200 genes with pure noise into Data A. Data C contains the expression levels of 1000 genes: adding another 500 pure noise genes into Data B. Data D contains the expression levels of 5000 genes: adding another 4000 pure noise genes into Data C. We also tried different sample size. Table 4.2 contains the result for one simulation with sample size 100, and Table 4.3 with sample size 20. Using PCA, genes with large loadings (for example, loading coefficients larger than half of the largest loading coefficient) will be picked up as possible biomarkers. For these four cases, the number of genes in the first five blocks contribute to the first three PCs are almost the same, as we can see in Table 4.2 and Table 4.3. More noisy genes included, we

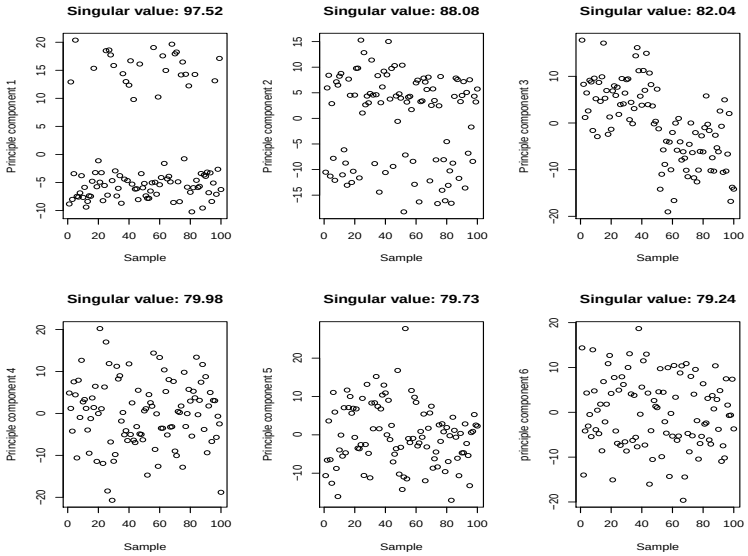

Figure 4.4: Ordinary principal component analysis of our simulated data: in addition to the first four blocks which contains some determinant structures, we add another 4700 rows of pure noise. Ordinary principal component analysis discovers 3 signals which are plotted in the first row. Obviously the third principal component is the specific signal we are looking for. It has a singular value of 82.04, which is just a little higher than those of the fourth and fifth principal components.

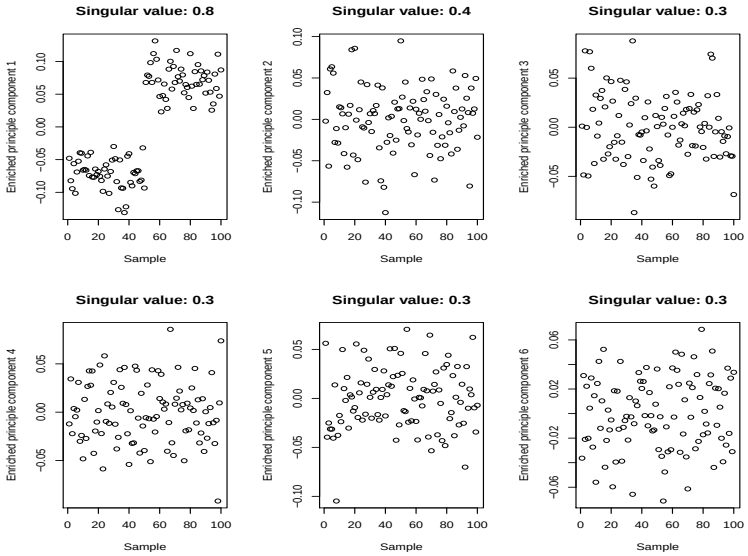

Figure 4.5: Enriched principal component analysis of the 5000×100 data matrix as in Figure 4.4. Through weighting, we strengthen the specific signal and weaken the others. The first principal component from enriched PCA is what we want. It has a singular value of 0.8 which is two times of the singular value of the second principal component.

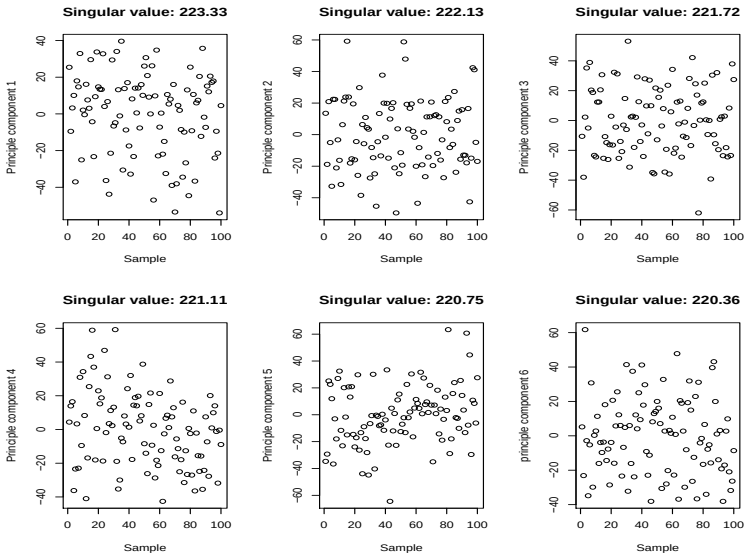

Figure 4.6: Ordinary principal component analysis of our simulated data: in addition to the first four blocks which contains some determinant structures, we add another 44700 rows of pure noise. The plot here shows the 44700 rows of pure noise conceal the determinant signal of the first four blocks. It seems the forth and fifth principal components show some correlation to the response, but not obviously.

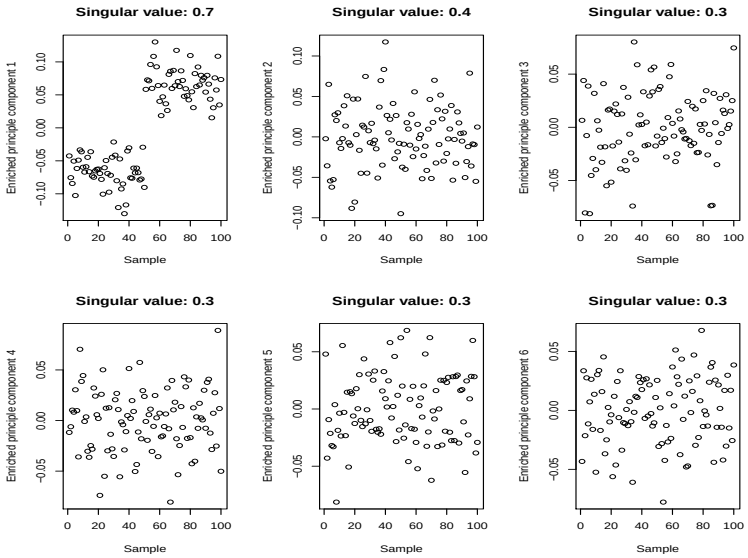

Figure 4.7: Enriched principal component analysis of the 45000×100 data matrix as in Figure 4.6. Enriched PCA is not affected by the huge number of noise. Still, it strengthens the specific signal. Though now we have a singular value of 0.7, instead of 0.8.

get more noisy genes contributed in the three PCs.

We also tried enriched PCA onto Data D, with conditional t test and Limma t test separately. Similarly, genes with large loadings (for example, loading coefficients larger than half of the largest loading coefficient) in our enriched PCA will be picked up as possible biomarkers. We can see that Limma t test is more conservative than conditional t test: while using conditional t test, 37 genes out of the first 50 genes are picked up, Limma t only picks 5 genes out of the first 50 genes, when sample size is 100 (shown in Table 4.2). Also in Table 4.3, when sample size is 20, conditional t enriched PCA picks 12 genes out of the first 50 genes, while Limma t only picks 5 genes.

For penalized methods (GLMNET and Penalized SVM) on Data D, adding L_2 norm to the penalty functions pick up more informative genes than enriched PCA, but with the price of a large proportion of noisy genes, comparing to the additional number of informative genes being picked. When the sample size is 20, noisy genes being picked up are around four times of informative genes that have been picked up.

The fitting errors and prediction errors of each method are stored in Table 4.1, where we can see that for ordinary PCA-LDA, the classification error also increase with the number of noisy genes. By Table 4.1, we see that our enriched PCA gives a smaller fitting error and prediction error, comparing to GLMNET and Penalized SVM. When sample size is smaller, i.e. 20, the advantage of our enriched method is more obvious.

	Sample size 100		Sample size 20	
Method	Fit Error	Prediction Error	Fit Error	Prediction Error
Ordinary PCA-LDA (A)	1	1	0	1
Ordinary PCA-LDA (B)	1	1	1	1
Ordinary PCA-LDA (C)	1	3	1	1
Ordinary PCA-LDA (D)	13	18	5	10
Enriched PCA-LDA Limma t	0	0	0	0
Enriched PCA-LDA conditional t	0	0	0	0
GLMnet(LASSO)	0	2.04	1.45	5.41
GLMnet(α=0.5)	0	1.00	0.75	5.44
SVM	0	13	0	6
SCAD SVM	0	4	1	3
Elastic SCAD SVM	0	1	0	4

Table 4.1: Compare the misclassification error of different methods. Simulation is based on model 4.1. The advantage of our enriched method is more obvious when the sample size is small, i.e., 20.

Genes selected in each block						
Method		Block 1	Block 2	Block 3	Block 4	Block 5
OPC	PC1	1	8	3	100	0
Data A	PC2	7	50	8	14	0
300 genes	PC3	50	13	24	13	0
	Total	50	50	33	100	0
OPC	PC1	4	11	4	100	5
Data B	PC2	8	50	8	15	14
500 genes	PC3	50	12	25	16	32
	Total	50	50	34	100	50
OPC	PC1	3	12	4	100	28
Data C	PC2	8	50	9	19	53
1000 genes	PC3	50	15	23	13	86
	Total	50	50	32	100	159
OPC	PC1	5	19	6	100	287
Data D	PC2	8	50	9	23	420
5000 genes	PC3	50	15	13	18	568
	Total	50	50	26	100	1168
EnrichedPCA	PC1	37	0	0	0	0
	PC2	23	0	0	0	0
Conditional t	Total	37	0	0	0	0
EnrichedPCA	PC1	1	0	0	0	0
	PC2	4	0	0	0	0
Limma t	Total	5	0	0	0	0
GLMNET	LASSO	29	0	0	0	6
	$\alpha = 0.5$	47	0	0	0	22
PenalizedSVM	SCAD	29	0	0	0	3
	SCAD+L_2	44	0	0	0	43

Table 4.2: Genes selected in each block by different methods. Simulation is based on model 4.1: 5000×100 data matrix. For Data A, B, C, D, ordinary principal component analysis picks almost the same number of genes from the first 4 blocks, and the number of noisy genes picked up increase with the number of noisy genes in the data. Enriched-PCA, GLMNET, Penalized SVM are applied on data D. We can see that Limma t is more conservative than conditional t. Penalized methods (GLMNET and Penalized SVM) including L_2 norm pick up relatively more genes than conditional t enriched PCA, with the price of more noisy genes.

Genes selected in each block						
Method		Block 1	Block 2	Block 3	Block 4	Block 5
OPC	PC1	14	20	19	94	0
Data A	PC2	20	48	39	34	0
300 Genes	PC3	39	20	35	34	0
	Total	49	49	68	98	0
OPC	PC1	18	25	27	96	40
Data B	PC2	23	48	35	32	69
500 Genes	PC3	39	26	42	36	66
	Total	48	49	74	98	123
OPC	PC1	19	23	25	97	145
Data C	PC2	19	48	35	30	218
1000 Genes	PC3	38	19	30	36	203
	Total	47	49	69	98	419
OPC	PC1	20	27	34	93	1312
Data D	PC2	17	44	38	36	1475
5000 Genes	PC3	28	34	21	41	1510
	Total	43	49	66	96	3098
EnrichedPCA	PC1	12	0	1	0	7
	PC2	7	0	0	0	3
Conditional t	Total	12	0	1	0	7
EnrichedPCA	PC1	2	0	0	0	0
	PC2	4	0	0	0	1
Limma t	Total	5	0	0	0	1
GLMNET	LASSO	5	0	0	0	8
	$\alpha = 0.5$	17	0	1	0	56
PenalizedSVM	SCAD	4	0	0	0	1
	SCAD+L_2	26	2	2	3	101

Table 4.3: Simulation using model 4.1: 5000×20 data matrix. This table is almost the same as Table 4.2, except that here the sample size is 20.

4.4 Analysis of our Data

Our method and some other popular methods will be evaluated on the dataset from one experiment for finding out the pathogenic mechanism of sialic acid storage diseases (which is due to the mutations in gene "Slc17A5" - encoding the protein Sialin). Raghavan et al. (2007) [53] and Amaratunga et al. (2008) [5] studied this data set. Two groups of mouse are studied in this experiment. One group consists of mice with gene "Slc17A5" knocked out, while the other group consists of wild type mice. Data is collected at three time points. At day 0, RNA samples from the whole embryo are studied. At this stage, there are no obvious phenotype traits that differ between the two groups but there would be gene expression differences. At day 10 and day 18, RNA samples from total brain tissue are studied. At these two stages, impaired sialic acid transport leads to observable morphological alterations such as defects in myelination. There are 6 biological samples for each group at each time point, except the control group for day 10, which has only 5 biological samples. All the samples (35 samples) are independent. For each sample, expression levels of 45101 genes are measured. In terms of sample size, the Sialin data is very typical, since microarray experiments are generally of sample size 3 to 10 per group. Some data sets generated by clinical trials are larger in sample size, but sometimes the samples are not independent and they usually come with covariates such as gender, race, age and others. Hence for all practical purposes the small sample size remains.

4.4.1 Our New Perspective

All the analysis that have been done about this data set till now treated each time point separately. Different models were built for different time points and were validated by cross validation or permutation on its own data sets. As we have noticed that, for each time point, the sample size is 12 or 11, which is quite small for cross validation. But as mice at each time point got the same treatment (half were wild mice, half received the treatment: gene "Slc17A5" was knocked out), these data were about their development at different time points. Intuitively, data on day 0 should give some information about what will happen on day 10 and day 18; and similarly, by studying the data on day 18, we could get some

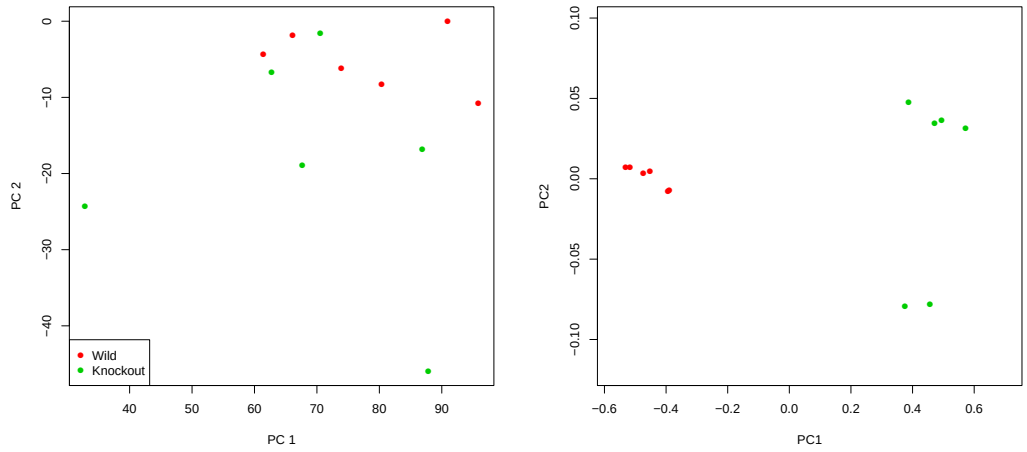

Figure 4.8: PC1 vs PC2 of day 0: standard PCA (left) and enriched PCA (right)

idea about what might happen on day 0 and day 10. Therefore, our new perspective about analyzing these three data is: after one model is built for some time point, data on the other two time points will serve to validate the model and the finding. In this way, we can avoid applying cross validation to data with small sample size. Moreover, we also gain more power by treating these three time points together.

4.4.2 Value of Enriching

Comparing to Ordinary Principal Component Analysis

In Figure 4.8, we display the results of performing PCA on the Sialin data of day zero (Sialin 0): both standard and enriched. The reason of emphasizing data of day 0 instead of day 10 or day 18 here is: the truly informative genes on day 0 is much less, comparing to other two time points, i.e., the ratio of noise to signal is pretty large at day 0; hence, the advantage of enriching is much more obvious. The left plot in Figure 4.8 shows the graph of the first two principal components from the standard PCA analysis. There is no separation between the knock-out group and the wild type group. The huge amount of noise conceal the specific signals. The right plot shows the first 2 principal components from the enriched PCA, and this time the separation is perfect. This shows clearly the value of enriching: strengthening the specific signal.

Comparing to Filtering Principal Component Analysis

Now let's see the advantage of weighting comparing to gene selection (filtering) based on day 18 data. The reason for emphasizing day 18 data now is: at day 18, a lot of things are going on in the brain; the structure of the data is very complicated; hence identifying the truly informative genes (biomarkers), based on which we could get the perfect predictions of day 0 and day 10, is more challenging than at the other two time points. The usual way to do filtering is to set a threshold for the p values and choose genes with p values smaller than that threshold. Then do analysis with the set of genes chosen. Here in this section, we show the plot of the first and second principal components of day 18 data in dots, the projections of data on day 0 (as described in section 4.2.1) in circles, and the projections of day 10 data in triangles. We tried thresholds 0.01, 0.05, and 0.1 respectively on the p values of conditional t test on data of day 18, picking up 2218, 4757, and 7306 genes respectively. The plots are shown in Figure 4.9, 4.10. In all figures, day 18 and day 10 data are well separated by some line, but not day 0 data.

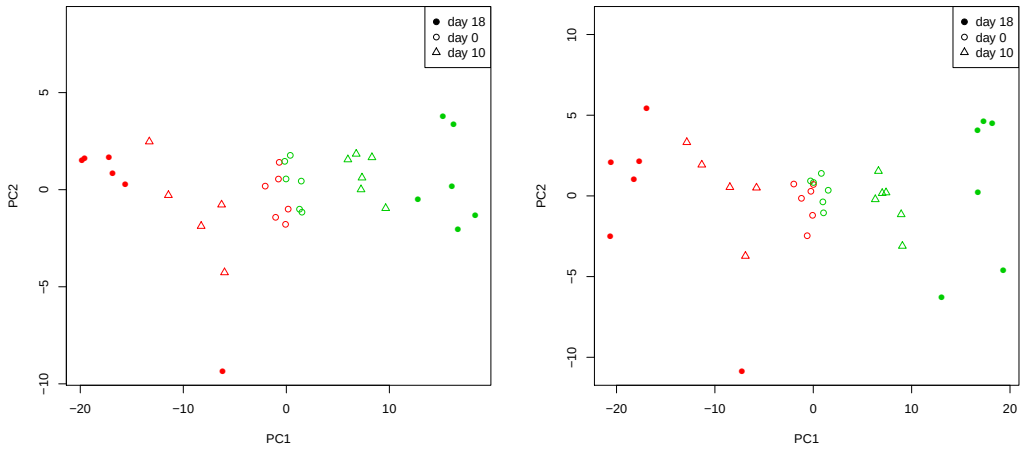

Figure 4.9: PCs 1 & 2; Filter out genes with p values larger than 0.01 (left) and 0.05 (right); day 18

Setting a threshold for p values can be criticized to be subjective. To improve, we can do some sophisticated filtering: let the data decide the threshold through cross validation. 3-fold and 6-fold cross validations are tried. With 3-fold cross validation, each time, $4 + 4$ samples are used to build one model and the number of errors in predicting the classes

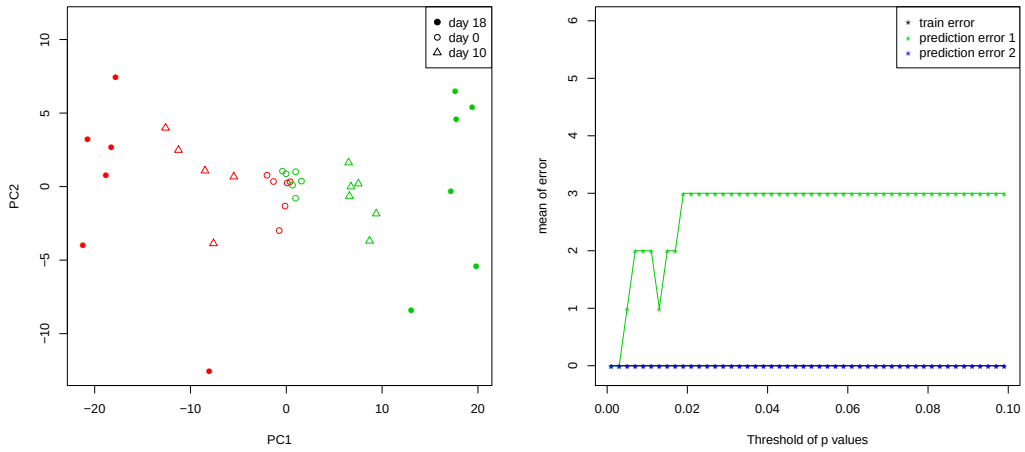

Figure 4.10: Left plot is about PCs 1 & 2; Filter out genes with q values larger than 0.1 of day 18. Right plot is about cross validation to determine the threshold for p value

of the remaining $2+2$ samples is calculated. With 6-fold cross validation, we build the model using $5+5$ samples and get prediction error for the remaining $1+1$ samples. Genes are selected by the order of significance based on p values. The mean of the number of prediction errors from 50 kinds of splitting is obtained. In predicting the day 0 and day 10 data, we use the whole day 18 data. Since any threshold between 0.001 to 0.5 for p value gives 0 test error for cross validation, 3-fold and 6-fold cross validation give similar plots of test error and prediction error: Figure 4.10. Prediction of day 10 data based on day 18 data is not challenging: the prediction error is always 0. The flat test error makes it impossible to choose the threshold for p value based on cross validation on day 18 data. However, the green curve shows that to get a perfect prediction of day 0 data, the threshold for filtering should not be bigger than 0.002 (obviously, the threshold should not be too small either). In all, by cross validation on day 18 data alone, any threshold from 0.001 to 0.5 works, which actually gives no insight about the threshold. Even if one has day 10 data in addition to day 18 data, still, any threshold from 0.001 to 0.5 works, as we have seen in Figure 4.10, the prediction error of day 10 based on day 18 is always 0. But to get perfect prediction of day 0 data, threshold should be in some interval (Figure 4.11). This makes the performance of filtering unstable.

Figure 4.11 is the plot for enriched method. We see that the performance of our enriched

method is perfect. To conclude, we have seen that with data of small sample size, filtering by subjectively thresholding or by cross validation is challenging and unstable. Our method bypasses the messy hard-filtering procedure, getting efficiency without losing any power.

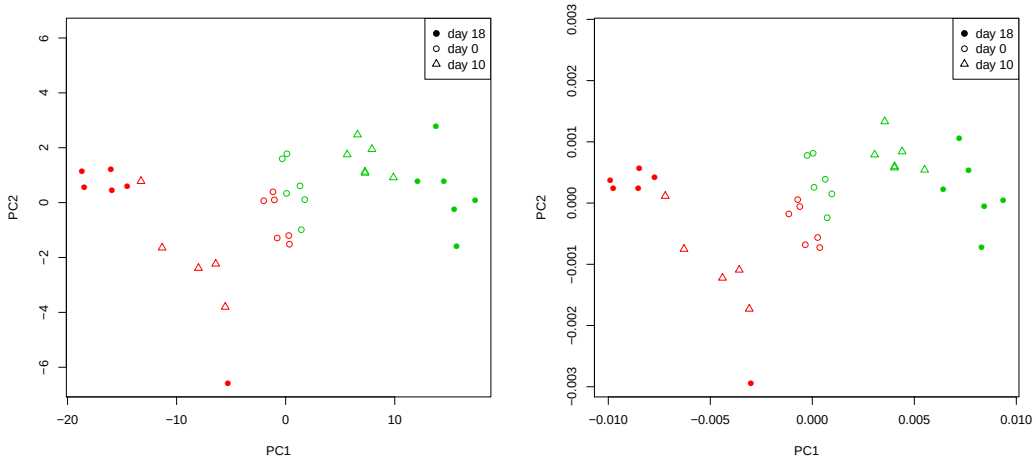

Figure 4.11: Left plot: PCs 1 & 2; Filter out genes with q values larger than 0.001; day 18. Right plot: PCs of day 18 by enriched PCA; project day 0 onto these two directions

4.4.3 Visualization of our Data by Enriched Biplot

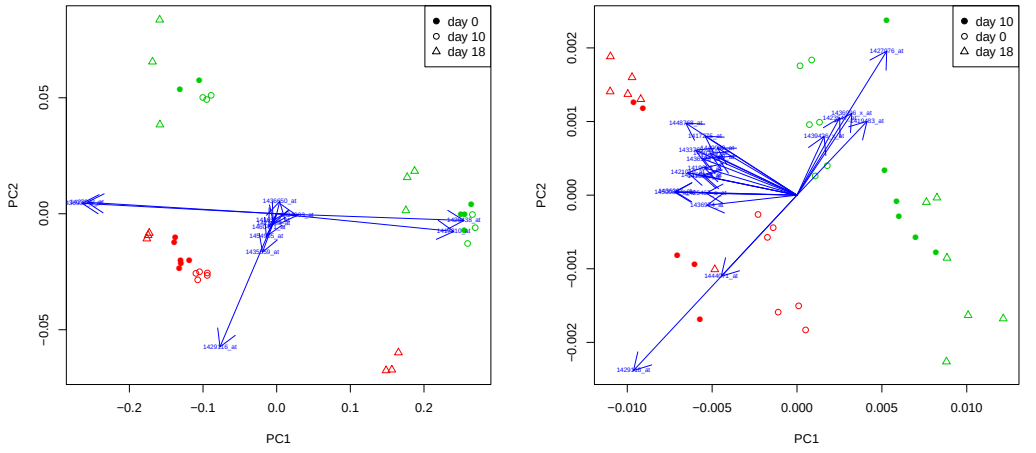

Figure 4.12: Left plot: Biplot of day 0; Right plot: Biplot of day 10.

Now we show the visualization of our data using weighted principle components analysis. In left plot of Figure 4.12, we plot the first two principal components of day 0 data in dots,

Genes 2nd PCA	*Annotation*
1429116_a	SLC gene that was knocked out in treatment group.
1435559_at	Myo6, Growth.
1437522_x_at	Gh growth hormone.
1454905_at	Inhibitor of Bruton agammaglobulinemia tyrosine kinase.
Genes 1st PCA	*Annotation*
1417210_at	Eukaryotic translation initiation factor 2, subunit 3, structural gene Y-linked Eif-2gy, Spy, Tfy
1426438_at	Box polypeptide 3, Y-link ed 8030469F12Rik, D1Pas1-rs1, Dby
1427262_at	Inactive X specific transcripts A430022B11, AI314753. Exper Embryonic brain development.
1436936_s_at	Inactive X specific transcripts A430022B11, AI314753 Experiment Embryonic brain development.

Table 4.4: Gene annotations for the two components of the enriched PCA of the Sialin day 0 data. Scientists could find interesting biomarkers from the loadings in the 2nd PCA.

projections of day 10 data in circles and projections of day 18 data in triangles. Similar plots for day 10 and day 18 are in Figure 4.12 and 4.13. In all plots, we visualize genes with relatively large contribution in classification in terms of their loadings in the principal components. Genes with loadings greater than some threshold will be shown as vectors in the plot. The lengths of these vectors in the plot are shrunk to the range of principal components.

Separation of Signals

As we have expected, the structures in data of day 0 (Figure 4.12) are relatively simpler than those of day 10 and day 18. Genes contribute in two main directions. The direction of the separating hyperplane, which is almost parallel to the second principal component, is dominated by genes: 1429116_a, 1435559_at, 1437522_x_at and 1454905_at. The other direction, which is almost parallel to the separating hyperplane and the first principal component, is dominated by genes: 1417210_at, 1426438_at, 1427262_at and 1436936_s_at.

In Table 4.4 we give a description of the annotations for each of the genes that contributes to the first and second principal components. It appears that the second principal component is specific and contains the Sialin gene plus other genes whose annotations indicate their contribution to growth pathways. This is in fact the type of signal that we are

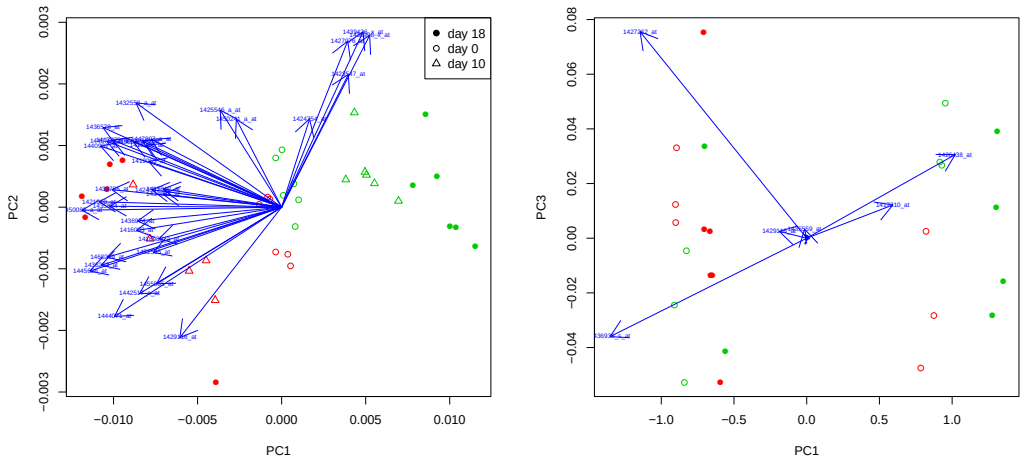

Figure 4.13: Left plot: Biplot of day 18. Right plot: PCs 1 3 of enriched PCA of day 0.

after and help us learn which pathways are affected by the knocked out gene at the embryo phase of the mouse development. The genes contribute to this principal component could be biomarkers. The first principal component is loaded with genes that are part of the X and Y chromosomes and therefore the second component corresponds to gender. This is what we call a non-specific signal. The fact that the two groups are not balanced with respect to the response implies that gender genes are differentially expressed and contribute to the signal.

This example helps us understand the three types of signals that appear in microarray data:

- The signal found in the second principal component is specific and is the primary signal of interest.
- The signal found in the first principal component is a non-specific signal that is attributable to a secondary effect in the data.
- The spurious signal in the third principal component (Figure 4.13), which does not give any information of day 0 data.

The enriched method is able to avoid the spurious signal and detect the other two. Then it is up to the scientist or analyst to separate the specific signals from the rest.

4.4.4 Comparing Results of all Methods in terms of Prediction Error

Here, we compare the prediction power of our method with some of the methods we reviewed in section 3.3. When cross validation is needed, we use 6-fold cross validation. When the performance of one method is not stable, we perform 100 times and calculate the mean number of prediction errors. Working with enriched PCA-LDA, we only choose the first 2 or 3 principle components. The SVM is performed with the R package e1071. We have tried the four kinds of penalty in the R package "penalizedSVM". The results are in Table 4.5.

The predictions of enriched PCA-LDA are perfect. Logistic model with L_1 and L_2 penalties works best when $\alpha = 1$, i.e. with L_1 penalty only. Enriched SVM does not work as well as we expected. The SVM model with L_1 and L_2 penalties (ElasticNet SVM) does not converge. Among the three penalties, SCAD SVM works best.

From Figure 4.12, to Figure 4.13, we can see that more and more genes become active. On day 0 and day 10, Sialin gene (1429116_at) has a large leverage in the direction of separating plane, but not on day 18. Since most genes shown in Figure 4.13 does not appear in Figure 4.12, prediction of day 0 data based on day 18 data is the most difficult one (see Table 4.5). Our enriched method handles it perfectly.

4.4.5 Conclusion

In this chapter, we introduced one efficient weighted principle component analysis method (we call enriched PCA-LDA), bypassing the subjectively thresholding step or the cross validation step in feature filtering methods, without losing any power. Enriched PCA-LDA helps separating different signals in the data, hence helps identifying the specific signals. Penalization can not separate the signals into specific and non-specific. After getting a subset of genes, one can not decide which gene contributes to the specific signal. Hence, enriched PCA-LDA improves the interpretation about the data. We compared the prediction results to some other popular methods, showing enriched PCA-LDA works perfectly for our data, even in the most difficult situation: predicting day 0 data from day 18 data.

Method	D0-10	D0-18	D10-0	D10-18	D18-0	D18-10
Enriched PCA-LDA Limma t	0	0	0	0	0	0
Enriched PCA-LDA conditional t	0	0	0	0	0	0
GLMnet(LASSO)	0.78	0.78	0.30	0.36	1.23	0.12
GLMnet(α=0.75)	1.10	0.88	6	0	6	0
GLMnet(α=0.5)	0.97	2.79	6	0	6	0
SVM	6	6	6	6	6	5
Enriched $(1/q)$ SVM	0	0	6	6	6	0
Enriched $(-\log)$ SVM	0	0	6	6	6	0
Proximal SVM	6	6	6	6	6	5
LASSO SVM	0	2	4	0	6	2
SCAD SVM	0	0	0	1	6	1
ElasticNet SVM	-	-	-	-	-	-
Elastic SCAD SVM	0	6	6	0	6	1

Table 4.5: Comparisons of the prediction performance of different methods. D0-10 means using day 0 data to predict day 10 data, D0-18 means using day 0 data to predict day 18 data and so on. Each cell is the number of prediction errors. '-' means the algorithm does not converge. The fraction number is the mean of prediction errors in 100 repeats.

The drawback of this method could be that the correlation between genes is ignored in the weighting step. As in microarray experiments, mRNA is extracted out of specific animals and tissues under biological conditions that are functionally associated with the mechanism examined, consequently, the genes tend to subgroup into highly correlated expression levels for reasons such as co-regulations based on genomic locations. In future work, we are going to consider the dependency structure among genes. But as we do not exclude genes completely, it could be all right in some situation, for example, data used in this chapter.

Chapter 5

Stochastic Approximation

5.1 Introduction

One characteristic of microarray data is that the number of variables (genes) is in the thousands while the sample size is only a few dozen. Statistical power is quite poor in analysis about each gene. Borrowing strength across genes is one popular way to improve the performance of such statistical analysis. In methods for borrowing strength across genes, the model used for the expression levels of gene g is generally assumed as follows:

$$\begin{aligned} X_g|\sigma_g &\sim N(\mu_g, \sigma_g^2) \\ \sigma_g &\sim F(\cdot). \end{aligned}$$

For example, in Limma t, $F(\cdot)$ assumes one inverse χ^2 distribution. In conditional t, effort is made to estimate $F(\cdot)$ from the empirical distribution of the sample variances of all genes. But because of the small sample size, the sample variances are poor estimates of the true variances for the genes, hence, the empirical distribution of the sample variances is a poor estimate of F. To improve the estimation, resampling methods are applied (Section 3.1.5).

In this chapter, we'd like to show the theoretical properties of the resampling methods used in conditional t test. Here are some notations which will be used in this chapter. Suppose we observe the expression levels of gene g in m samples: $X_g = \{x_{g1}, x_{g2}, \cdots, x_{gm}\}$. We estimate σ_g^2 by the sample variance of X_g: s_g^2. Then the empirical distribution of $\{s_g^2 : g = 1, \cdots, G\}$ ($\hat{F}_s$) will be used to estimate F.

The idea used in conditional t is assuming there is a mapping:

$$\mathbf{M} : F \in \mathcal{F} \to \hat{F}_s \in \mathcal{F}_G,$$

where $\mathcal{F}$ is the set of all possible probability distributions and $\mathcal{F}_G$ is the set of all step

probability distribution. For the $\hat{F}_s$ we get, we hope to recover the true F by estimating the mapping $\mathbf{M}$, thus improve the performance of $\hat{F}_s$.

5.2 Preliminary

Here we review some methods in similar problems, which will give us some idea about how to improve the performance of $\hat{F}_s$.

5.2.1 Stochastic Approximation

Robbins and Monro (1951) [55] proposed stochastic approximation to solve the equation

$$M(x) = \alpha, \tag{5.1}$$

where $M(x) = \int_{-\infty}^{\infty} y dH(y|x)$ is the expected value of Y for the given x. They assumed:

- $H(y|x)$ is, for every x, a distribution function in y, and that there exists a positive constant C such that

$$Pr[|Y(x)| \leq C] = \int_{-C}^{C} dH(y|x) = 1, \quad \forall x. \tag{5.2}$$

- For every x the expected value $M(x)$ exists and is finite.

- There exist finite constants α, θ such that

$$\begin{aligned} M(x) &\leq \alpha \text{ for } x < \theta, \\ M(x) &\geq \alpha \text{ for } x > \theta. \end{aligned} \tag{5.3}$$

They defined a (nonstationary) Markov chain $\{x_n\}$ by taking x_1 to be arbitrary constant and defining

$$x_{n+1} - x_n = a_n(\alpha - y_n), \tag{5.4}$$

where y_n is a random variable such that $Pr[y_n \leq y|x_n] = H(y|x_n)$, and $\{a_n\}$ is a fixed sequence of positive constants such that $0 < \sum_1^{\infty} a_n^2 = A < \infty$. They showed under some conditions $x_n \to \theta$ in probability.

Wolfowitz (1952) [74] weakened conditions in Robbins and Monro (1951) [55] and assumed $|M(x)| \leq C < \infty$, $\int_{-\infty}^{\infty}(y - M(x))^2 dH(y|x) \leq \sigma^2 < \infty$, and that either $M(x) \leq \alpha - \delta,\ x < \theta;\ M(x) \geq \alpha + \delta,\ x > \theta$, for some $\delta > 0$, or else $M(x) < \alpha$, for $x < \theta;\ M(\theta) = \alpha;\ M(x) > \alpha$, for $x > \theta$, and for some positive δ, $M(x)$ is strictly increasing if $|x - \theta| < \delta$ and $\inf_{|x-\theta|\geq\delta} |M(x) - \alpha| > 0$. Let $\{a_n\}$ be a sequence of positive numbers such that $\sum_{n=1}^{\infty} a_n = \infty$, $\sum_{n=1}^{\infty} a_n^2 < \infty$. Let x_1 be an arbitrary number. The Robbins-Monro convergence scheme is defined recursively for all n by $x_{n+1} = x_n + a_n(\alpha - y_n)$, where y_n is a random variable with distribution function $H(y|x_n)$. They showed that x_n converges to θ in probability.

Blum (1954) [14] extended to multidimensional cases, assuming $\{Y^{(1)}_{x_1,\cdots,x_k}\}, \cdots, \{Y^{(k)}_{x_1,\cdots,x_k}\}$ are k families of random variables with corresponding families of distribution functions $\{F^{(1)}_{x_1,\cdots,x_k}\}, \cdots, \{F^{(k)}_{x_1,\cdots,x_k}\}$, each depending on k real variables $(x_1, \cdots, x_k)$. Let

$$M^{(i)}(x_1, \cdots, x_k) = \int_{-\infty}^{\infty} y dF^{(i)}_{x_1,\cdots,x_k},$$

for $i = 1, \cdots, k$, be the corresponding regression functions. Assume that it is possible to make an observation on the random variable $Y^{(i)}_{x_1,\cdots,x_k}$ for $i = 1, \cdots, k$, and any choice of real numbers $(x_1, \cdots, x_k)$. If $\alpha_1, \cdots, \alpha_k$ are k specified numbers,

$$M^{(i)}(x_1, \cdots, x_k) = \alpha_i, \qquad i = 1, \cdots, k,$$

Blum created one sequence of approximating random vectors which converges a.s. to a solution of the equation.

There are some other papers about Robbins-Monro stochastic approximation procedure (RM procedure). Blum (1954) [15] and Dvoretzky (1956) [27] showed that under some conditions the RM procedure would converge almost surely. Sacks (1958) [57] showed that under some conditions, the RM procedure would converge in distribution.

One assumption made by Robbins-Monro stochastic approximation is that we can get observations sequentially, which is not the case in experiments about microarray technology. Because the time and cost involved, the number of observations in microarray data is often small and fixed.

5.2.2 Double Bootstrap

Hall and Martin (1988) [39] and Martin (1992) [51] proposed a unifying approach to a general form of bootstrap resampling. Let F_0 be the true distribution, F_1 be the empirical distribution based on a random sample, $f(F_0, F_1)$ be a known function of F_0 and F_1, and $E(\cdot|F_0)$ be expectation given that the data came from F_0. They pointed out that a typical statistical problem has the form: choose f from a designated class so that

$$E\{f(F_0, F_1)|F_0\} = 0. \tag{5.5}$$

The resampling solution of f runs as follows: draw a same-size sample at random, with replacement, from the original sample; let F_2 denote its empirical distribution function, conditional on F_1; arguing that the relationship between F_1 and F_2 should be similar to that between F_0 and F_1, choose f from the class of eligible f's so that

$$E\{f(F_1, F_2)|F_1\} = 0. \tag{5.6}$$

The resulting f, depending on F_1, approximates the unobservable f which solves (5.5). If the problem could be formulated in a way that f could be indexed by a vector $\mathbf{t}$, and take $T(F_0)$ to be the 'ideal' $\mathbf{t}$ which solves (5.5), the resampling principle produces $T(F_1)$, the solution of (5.6), as the bootstrap estimate of $T(F_0)$. Then $f_{T(F_1)}$ is the bootstrap estimate of f.

The idea behind the iterated bootstrap is a simple but powerful one: use the bootstrap to estimate and correct for errors in bootstrap procedures. This involves resampling from samples that are themselves resamples from the original data; that is, bootstrap the bootstrap.

Suppose one is interested in reducing the bias of an estimator $\hat{\theta} = \psi(F_1)$ in estimating $\theta = \psi(F_0)$. The bias of $\hat{\theta}$ in estimating θ is

$$\text{bias}(\hat{\theta}) = E(\hat{\theta}) - \theta = E(\psi(F_1)|F_0) - \psi(F_0).$$

The bootstrap estimate of the bias of $\hat{\theta}$ is constructed as follows. Let $\mathbf{X}^\star$ be a resample from $\mathbf{X}$, and denote by $\hat{\theta}^\star$ the version of $\hat{\theta}$ computed using the resample $\mathbf{X}^\star$ instead of the sample $\mathbf{X}$. The bootstrap estimate of bias is

$$\widehat{\text{bias}}(\hat{\theta}) = E(\hat{\theta}^\star|\mathbf{X}) - \hat{\theta} = E(\psi(F_2)|F_1) - \psi(F_1),$$

where F_2 is the empirical distribution, conditional on F_1, based on the resample $\mathbf{X}^\star$. Consequently, the bootstrap bias-corrected estimate of θ is

$$\hat{\theta}_1 = \hat{\theta} - \widehat{\text{bias}}(\hat{\theta}) = 2\psi(F_1) - E(\psi(F_2)|F_1).$$

Bias can be further reduced by iterating the above procedure. The form of the estimator after j bias corrections is:

$$\hat{\theta}_j = \sum_{i=1}^{j+1} C_{j+1}^i (-1)^{i+1} E\{\psi(F_i)|F_1\},$$

where F_i is the empirical distribution, conditional on F_{i-1}, of a resample from F_{i-1}. The bootstrap quantity $E\{\psi(F_i)|F_1\}$ is approximated by Monte Carlo algorithm. The gave asymptotic results about the orders of bias. However, when sample sizes are small, high-order iterations can result in increased bias.

The idea 'bootstrap the bootstrap' is quite appealing for microarray data, since we could not get more observations most of the time. But then it is difficult to formulate our problem in a form like (5.5).

5.2.3 Target Estimation

Cabrera and Fernholz (1999) [21] proposed target estimation to correct the bias of some estimator. Suppose the statistic $T(F_n)$ estimates the parameter $T(F_\theta)$, where T is a statistical functional and F_n is the empirical distribution function corresponding to the sample $X_1, \cdots, X_n$ (i.i.d. random variables with common distribution function F_θ, with $\theta \in \Theta$). Assume the expectation of $T(F_n)$, $g(\theta) = E_\theta(T(F_n))$, exists for all $\theta \in \Theta$, where E_θ indicates the expectation with respect to F_θ. The function g is assumed to be one-to-one and differentiable.

$$g^{-1}(T) =: \widetilde{T}$$

is called the the target functional of T and $\widetilde{T}(F_n)$ will be the target estimator of θ.

The target estimate of θ corresponds to choosing the value $\theta = \widetilde{T}(\hat{F}_n)$, which solves the equation

$$E_\theta(T(F_n)) = T(\hat{F}_n),$$

where $\hat{F}_n$ is the observed value of F_n.

They showed that when $g(\theta)$ satisfies some properties, targeting will reduce the bias, and when the von Mises kernels of T satisfy some properties, the variance of the target estimator will be reduced.

Cabrera and Hu (2001) [22] showed that if $g(\theta) = \theta + b(\theta)n^{-1} + O(n^{-2})$, $E_\theta(T_n - \theta)^2 = O(n^{-1})$ then

$$[1 + b'(\theta)n^{-1}] \cdot E_\theta[g^{-1}(T_n) - \theta] = O(n^{-2}),$$

which shows that the target estimator reduces the bias of a root n consistent estimator T_n by a factor of n. Using the target estimation approach, Cabrera and Watson (1997) [24] introduced a median bias reduction procedure. Fernholz (1997) [29] studied the asymptotic normality and the robustness of target estimator. Cabrera and Hu (2001) [22] also suggested that using Robbins and Monro's stochastic approximation to implement the target estimation. They compared their algorithm with Bootstrap and Jackknife bias reduction methods by simulation and demonstrated that target estimation is quite useful. Cabrera et al. (2005) [20] applied target estimation to logistic regression models. Cabrera and Fernholz (2004) [23] extended the target estimation to multivariate situations.

The way Cabrera and Fernholz (1999) [21] formulated the problem gives us the idea about how to formulate our problem.

5.3 Stochastic Approximation to Improve $\hat{F}_s$

We formulate the problem in the following form:

$$\mathbf{M} : F \in \mathcal{F} \to \hat{F}_s \in \mathcal{F}_G,$$

and try to improve it by the idea of target estimation. Bootstrap the bootstrap is the main idea of the algorithm and each time we correct the estimation by a small step as in Robbins-Monro stochastic approximation. For the $\hat{F}_s$ we get, we hope to recover the true F to some extent, hence to improve $\hat{F}_s$.

Instead of working on continuous functions, let's work with the quantiles. Denote $q(\boldsymbol{\alpha})$ be the vector of quantiles corresponding to $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_G) = (1, 2, \cdots, G)/(G + 1)$. Reformulate the problem as

$$\mathbf{M}(q(\boldsymbol{\alpha})) = \hat{q}(\boldsymbol{\alpha}). \tag{5.7}$$

What kind of properties do $\mathbf{M}$ have? Obviously, no monotonicity exists, since the space $\mathcal{F}$ is partially ordered. It is also possible that $\mathbf{M}$ is unidentifiable. But we assume it is identifiable.

5.3.1 One Procedure

The procedure we propose is the following:

$$q^{n+1}(\boldsymbol{\alpha}) = \text{sort}\left\{q^n(\boldsymbol{\alpha}) + a_n(\hat{q}(\boldsymbol{\alpha}) - \mathbf{M}(q^n(\boldsymbol{\alpha})))\right\},$$

where sorting is trying to keep $q^{n+1}(\boldsymbol{\alpha})$ in increasing order. We choose $q^1(\boldsymbol{\alpha}) = \hat{q}(\boldsymbol{\alpha})$ and $a_n = \frac{1}{n}$ so that $\sum_{n=1}^{\infty} a_n = \infty$ and $\sum_{n=1}^{\infty} a_n^2 < \infty$ (the same as the procedure in Robbins and Monro (1995)).

We don't know $\mathbf{M}(\cdot)$, so we estimate it by bootstrap. For a given $q^n(\alpha)$, we simulate m samples from Normal$(0, q^n(\alpha_i))$ and get the sample variances s_i^2 $(i = 1, 2, \cdots, G)$.

$$\begin{aligned}
q^n(\alpha_1) &\rightarrow (X_{11}, X_{12}, \cdots, X_{1m}) \rightarrow s_1^2;\\
q^n(\alpha_2) &\rightarrow (X_{21}, X_{22}, \cdots, X_{2m}) \rightarrow s_2^2;\\
&\vdots\\
q^n(\alpha_G) &\rightarrow (X_{N1}, X_{N2}, \cdots, X_{Nm}) \rightarrow s_G^2.
\end{aligned}$$

Then

$$\mathbf{M}(q^n(\boldsymbol{\alpha})) = \text{sort}(s_1^2, s_2^2, \cdots, s_G^2) = s^n(\boldsymbol{\alpha}).$$

To improve the efficiency of the procedure, we may do the bootstrap B times, and get:

$$\mathbf{M}(q^n(\boldsymbol{\alpha})) = \text{mean}(\text{sort}(s_{b,1}^2, s_{b,2}^2, \cdots, s_{b,G}^2) : b = 1, \cdots, B).$$

By the simulation step, $(s_1^2, s_2^2, \cdots, s_G^2)$ are independent. As we assume that given the σ^2's

the X's follow normal distribution, we have

$$\left.\begin{pmatrix} s_1^2 \\ s_2^2 \\ \vdots \\ s_G^2 \end{pmatrix}\right| \left.\begin{pmatrix} \sigma_1^2 \\ \sigma_2^2 \\ \vdots \\ \sigma_G^2 \end{pmatrix}\right. \sim \frac{(m-1)^G}{\sigma_1^2\sigma_2^2\cdots\sigma_G^2} \cdot f(\frac{(m-1)s_1^2}{\sigma_1^2}) f(\frac{(m-1)s_2^2}{\sigma_2^2}) \cdots f(\frac{(m-1)s_G^2}{\sigma_G^2})$$

where f is the density function of χ^2_{m-1} distribution. To show the theoretical properties of our procedure, the perfect way will be getting the distribution of

$$\left.\begin{pmatrix} s_{(1)}^2 \\ s_{(2)}^2 \\ \vdots \\ s_{(G)}^2 \end{pmatrix}\right| \left.\begin{pmatrix} \sigma_{(1)}^2 \\ \sigma_{(2)}^2 \\ \vdots \\ \sigma_{(G)}^2 \end{pmatrix}\right. .$$

However, it seems impossible. Even for $G = 2$, the distribution function is quite messy and it is difficult to get the mean of $(s^2_{(1)}, s^2_{(2)})$. Instead of working on the high-dimensional vector, assumptions about each element $s^n(\alpha)|q^n(\alpha)$ are made. This is not exactly what we would love to prove, but is what we can do for the moment.

5.3.2 Assumption I

$$s^n(\alpha)|q^n(\alpha) \sim (q^n(\alpha), \frac{1}{m}). \tag{5.8}$$

Justification of this Assumption

This assumption is based on the following fact. Let $X_1, \cdots, X_m \sim F$. Fix k numbers: $0 < \alpha_1 < \alpha_2 < \cdots < \alpha_k < 1$. If F has density function f and f is positive and continuous

at the α_ith quantile $\xi_i (i = 1, \cdots, k)$ of F, denote:

$$\hat{\xi}_m = (\hat{\xi}_{m\alpha_1}, \cdots, \hat{\xi}_{m\alpha_k}), \qquad \xi = (\xi_{\alpha_1}, \cdots, \xi_{\alpha_k}),$$

where $\hat{\xi}_{m\alpha_i}$ is the sample α_ith quantile; then when $m \to \infty$, we have

$$\sqrt{m}(\hat{\xi}_m - \xi) \to N(0, \Lambda),$$

where Λ is a $k \times k$ matrix and its (i, j) element is

$$\lambda_{ij} = \alpha_i(1 - \alpha_j)/[f(\xi_i)f(\xi_j)].$$

Especially when $k = 1$,

$$\sqrt{m}(\hat{\xi}_{mp} - \xi_p) \to N(0, p(1 - p)/f^2(\xi_p)).$$

Thus, we see that $E(s^n(\alpha)|q^n(\alpha)) \sim q^n(\alpha)$ and $\mathrm{Var}(s^n(\alpha)|q^n(\alpha)) \sim O(1/m)$.

5.3.3 Assumption II

$$s^n(\alpha)|q^n(\alpha) \sim (q^n(\alpha), \frac{(q^n(\alpha))^2}{m}). \tag{5.9}$$

Justification of this Assumption

This assumption is based on the following fact. Suppose the samples are taken from a normal distribution. Then using the fact that $\frac{(m-1)S^2}{\sigma^2}$ is a chi squared random variable with $(m - 1)$ degrees of freedom, we get

$$\mathrm{Var}\,\frac{(m-1)S^2}{\sigma^2} = \mathrm{Var}\,\chi^2_{m-1} = 2(m - 1);$$

Hence:

$$\mathrm{Var}\,S^2 = \frac{2\sigma^4}{m-1}.$$

5.4 Convergence of the Procedure

Now we prove the convergence of our procedure under those two assumptions separately. we will show that that $q^n(\alpha)$ converges to the solution of (5.7): $q^\star(\alpha)$ in probability.

$$\begin{aligned} E(q^{n+1}(\alpha) - q^\star(\alpha))^2 &= (q^n(\alpha) - q^\star(\alpha) + a_n(\hat{q}(\alpha) - M(q^n(\alpha))))^2 \\ &= E(q^n(\alpha) - q^\star(\alpha))^2 + a_n^2 E(\hat{q}(\alpha) - M(q^n(\alpha)))^2 \\ &\quad +2a_n E[(q^n(\alpha) - q^\star(\alpha)) \cdot (\hat{q}(\alpha) - M(q^n(\alpha)))] \end{aligned}$$

Denote $\tau_n^2 = E(q^n(\alpha) - q^\star(\alpha))^2$. To prove the convergence of our procedure, we need to show $\tau_n^2 \to 0$.

$$\tau_{n+1}^2 = \tau_n^2 + a_n^2 E(\hat{q}(\alpha) - M(q^n(\alpha)))^2 + 2a_n E[(q^n(\alpha) - q^\star(\alpha)) \cdot (\hat{q}(\alpha) - M(q^n(\alpha)))]$$

When we use $s^n(\alpha)$ to estimate $M(q^n(\alpha))$, we have

$$\tau_{n+1}^2 = \tau_n^2 + a_n^2 E(\hat{q}(\alpha) - s^n(\alpha))^2 + 2a_n E[(q^n(\alpha) - q^\star(\alpha))(\hat{q}(\alpha) - s^n(\alpha))] \quad (5.10)$$

5.4.1 Under Assumption I

By assumption (5.8), we have:

$$\begin{aligned} E[(q^n(\alpha) - q^\star(\alpha)) \cdot (\hat{q}(\alpha) - s^n(\alpha))] &= E\{[(q^n(\alpha) - q^\star(\alpha)) \cdot (\hat{q}(\alpha) - s^n(\alpha))]|q^n(\alpha)\} \\ &= E[(q^n(\alpha) - q^\star(\alpha)) \cdot (\hat{q}(\alpha) - q^n(\alpha))] \quad (5.11) \end{aligned}$$

$$\begin{aligned} E(\hat{q}(\alpha) - s^n(\alpha))^2 &= \hat{q}(\alpha)^2 - 2\hat{q}(\alpha)Eq^n(\alpha) + E(s^n(\alpha))^2 \\ &= \hat{q}(\alpha)^2 - 2\hat{q}(\alpha)Eq^n(\alpha) + E(q^n(\alpha))^2 + \frac{1}{m} \\ &= E(\hat{q}(\alpha) - q^n(\alpha))^2 + \frac{1}{m} \quad (5.12) \end{aligned}$$

Plugging in (5.11) and (5.12) into (5.10), we have:

$$\begin{aligned} \tau_{n+1}^2 &= \tau_n^2 + a_n^2 E(\hat{q}(\alpha) - q^n(\alpha))^2 + 2a_n E[(q^n(\alpha) - q^\star(\alpha)) \cdot (\hat{q}(\alpha) - q^n(\alpha))] + a_n^2 \frac{1}{m} \\ &= (1 - a_n)^2 \tau_n^2 + 2a_n(1 - a_n)(\hat{q}(\alpha) - q^\star(\alpha))E(q^n(\alpha) - q^\star(\alpha)) \\ &\quad +a_n^2[\frac{1}{m} + (\hat{q}(\alpha) - q^\star(\alpha))^2] \quad (5.13) \end{aligned}$$

Now we need to check the term $E(q^n(\alpha) - q^\star(\alpha))$ in (5.13).

$$q^{n+1}(\alpha) - q^\star(\alpha) = q^n(\alpha) - q^\star(\alpha) + a_n(\hat{q}(\alpha) - s^n(\alpha)).$$

Denoting $b_n = E(q^n(\alpha) - q^\star(\alpha))$, as $a_n = \frac{1}{n}$, we have:

$$\begin{aligned} b_{n+1} &= (1-a_n)b_n + a_n(\hat{q}(\alpha) - q^\star(\alpha)) \\ &= (1-a_n)(1-a_{n-1})\cdot(1-a_2)b_2 + \\ &\quad [(1-a_n)(1-a_{n-1})\cdots(1-a_3)a_2 + \cdots + (1-a_n)a_{n-1} + a_n](\hat{q}(\alpha) - q^\star(\alpha)) \\ &= \frac{1}{n}b_2 + \frac{n-1}{n}(\hat{q}(\alpha) - q^\star(\alpha)) \end{aligned} \tag{5.14}$$

Plugging (5.14) into (5.13), we have:

$$\begin{aligned} \tau_{n+1}^2 &= (1-a_n)^2\tau_n^2 + 2a_n(1-a_n)(\hat{q}(\alpha) - q^\star(\alpha))E(q^n(\alpha) - q^\star(\alpha)) \\ &\qquad + a_n^2[\frac{1}{m} + (\hat{q}(\alpha) - q^\star(\alpha))^2] \\ &\approx (\frac{n-1}{n})^2\tau_n^2 + \frac{1}{n^2}C \end{aligned}$$

where $C = 2b_2C_0 + 2C_0^2 + \frac{1}{m}$ and $C_0 = \hat{q}(\alpha) - q^\star(\alpha)$.

Hence we have

$$\tau_{n+1}^2 = (\frac{3}{n})^2\tau_2^2 + \frac{n-3}{n^2}C$$

which says: $\tau_n^2 \to 0$. Therefore, our procedure converges under assumption I.

5.4.2 Under Assumption II

Continuing with (5.10):

$$\tau_{n+1}^2 = \tau_n^2 + a_n^2E(\hat{q}(\alpha) - s^n(\alpha))^2 + 2a_nE[(q^n(\alpha) - q^\star(\alpha))\cdot(\hat{q}(\alpha) - s^n(\alpha))].$$

By assumption (5.9), we have:

$$\begin{aligned} E(\hat{q}(\alpha) - s^n(\alpha))^2 &= \hat{q}(\alpha)^2 - 2\hat{q}(\alpha)Eq^n(\alpha) + E(s^n(\alpha))^2 \\ &= \hat{q}(\alpha)^2 - 2\hat{q}(\alpha)Eq^n(\alpha) + \frac{m+1}{m}E(q^n(\alpha))^2 \\ &= \frac{m+1}{m}E(q^n(\alpha) - \frac{m}{m+1}\hat{q}(\alpha))^2 + \frac{1}{m+1}(\hat{q}(\alpha))^2 \end{aligned}$$

Then

$$\begin{aligned}
\tau_{n+1}^2 &= \tau_n^2 + a_n^2 E(\hat{q}(\alpha) - s^n(\alpha))^2 + 2a_n E[(q^n(\alpha) - q^\star(\alpha)) \cdot (\hat{q}(\alpha) - s^n(\alpha))] \\
&= \tau_n^2 + a_n^2 \frac{m+1}{m} E\left(q^n(\alpha) - \frac{m}{m+1}\hat{q}(\alpha)\right)^2 + a_n^2 \frac{1}{m+1}(\hat{q}(\alpha))^2 + \\
&\quad 2a_n E[(q^n(\alpha) - q^\star(\alpha)) \cdot (\hat{q}(\alpha) - q^n(\alpha))] \\
&= \tau_n^2 + a_n^2 \frac{m+1}{m}\left[\tau_n^2 + C_1^2 + 2C_1 E(q^n(\alpha) - q^\star(\alpha))\right] + a_n^2 \frac{1}{m+1}(\hat{q}(\alpha))^2 \\
&\quad -2a_n\tau_n^2 + 2a_n C_0 E(q^n(\alpha) - q^\star(\alpha)) \\
&= \left[1 - 2a_n + \frac{m+1}{m}a_n^2\right]\tau_n^2 + \left[2a_n^2\frac{m+1}{m}C_1 + 2a_n C_0\right] b_n \\
&\quad + a_n^2\left[\frac{m+1}{m}C_1^2 + \frac{\hat{q}(\alpha)^2}{m+1}\right].
\end{aligned}$$

Since

$$\begin{aligned}
b_{n+1} &= (1-a_n)b_n + a_n(\hat{q}(\alpha) - q^\star(\alpha)) \\
&= (1-a_n)(1-a_{n-1}) \cdot (1-a_2)b_2 + \\
&\quad [(1-a_n)(1-a_{n-1}) \cdots (1-a_3)a_2 + \cdots + (1-a_n)a_{n-1} + a_n](\hat{q}(\alpha) - q^\star(\alpha)) \\
&= \frac{1}{n}b_2 + \frac{n-1}{n}(\hat{q}(\alpha) - q^\star(\alpha)),
\end{aligned}$$

we have:

$$\begin{aligned}
\tau_{n+1}^2 &= \left[1 - 2a_n + \frac{m+1}{m}a_n^2\right]\tau_n^2 + \left[2a_n^2\frac{m+1}{m}C_1 + 2a_n C_0\right] \cdot \left[\frac{1}{n}b_2 + \frac{n-1}{n}(\hat{q}(\alpha) - q^\star(\alpha))\right] \\
&\quad + a_n^2\left[\frac{m+1}{m}C_1^2 + \frac{\hat{q}(\alpha)^2}{m+1}\right] \\
&\leq \frac{m+1}{m}(1-a_n)^2\tau_n^2 + a_n^2 C_2 \\
&\leq (1-a_n)\tau_n^2 + a_n^2 C_2
\end{aligned}$$

The last inequality holds when $n \geq m+1$. Thus we have:

$$\begin{aligned}
\tau_{n+1}^2 &\leq (1-a_n)(1-a_{n-1})\tau_{n-1}^2 + [(1-a_n)a_{n-1}^2 + a_n^2]C_2 \\
&\leq (1-a_n) \cdots (1-a_{m+1})\tau_{m+1}^2 + \\
&\quad \left[(1-a_n) \cdots (1-a_{m+2})a_{m+1}^2 + \cdots + (1-a_n)a_{n-1}^2 + a_n^2\right] C_2 \\
&= \frac{m}{n}\tau_{m+1}^2 + \frac{1}{n}[\frac{1}{m+1} + \cdots + \frac{1}{n}]C_2 \\
&\leq \frac{m}{n}\tau_{m+1}^2 + \frac{\log n}{n}C_2 \\
&\to 0 \quad (n \to \infty)
\end{aligned}$$

Therefore, the procedure converges under assumption II.

5.5 Simulation

5.5.1 Estimating the Distribution of Variance

The model we assume is:

$$\begin{aligned} X|\sigma^2 &\sim N(0,\sigma^2) \\ \sigma^2 &\sim \chi^2_k. \end{aligned}$$

In our simulation, we have a sample of σ^2 of size 5000 from χ^2_k distribution; for each σ^2, we get a sample of size m from the distribution $N(0,\sigma^2)$.

Figure 5.1 is the simulation results about the situation when the sample size is 20 and the χ^2 distribution of the variances has 20 degrees of freedom. These plots are the quantiles of true variances versus the quantiles of estimated variances. The top left one is about the sample variances. The top right one is about the stochastic approximation with 5 iterations. The bottom left one is the stochastic approximation with 10 iterations and the bottom right one with 20 iterations. From the top left plot, we see that the distribution of sample variances is one inflated estimate of the distribution of true variances: the sample quantiles tend to underestimate the small quantiles of the true variances and overestimate the large quantiles of the true variances. We also see 5 iterations of our procedure will correct the bias obviously and 20 iterations will give a quite good estimation about the true distribution of the variances.

Figure 5.2 is the simulation results about the situation when the samples size is 20 and the χ^2 distribution of the variances has 10 degrees of freedom. In Figure 5.3, the sample size is 20 and the degree of freedom for the χ^2 distribution is 30. By comparing these 3 sets of figures, we see that when the sample size is smaller than the degrees of freedom (Figure 5.3), the empirical distributions of the sample variances behave quite badly. However, stochastic approximation provides good improvement over the sample variances. When the sample size is larger than the degrees of freedom, the empirical distribution of the sample variances

is acceptable (Figure 5.1). Of course, stochastic approximation works almost perfectly for this situation. Therefore, by simulation, we see the power of the stochastic approximation we proposed.

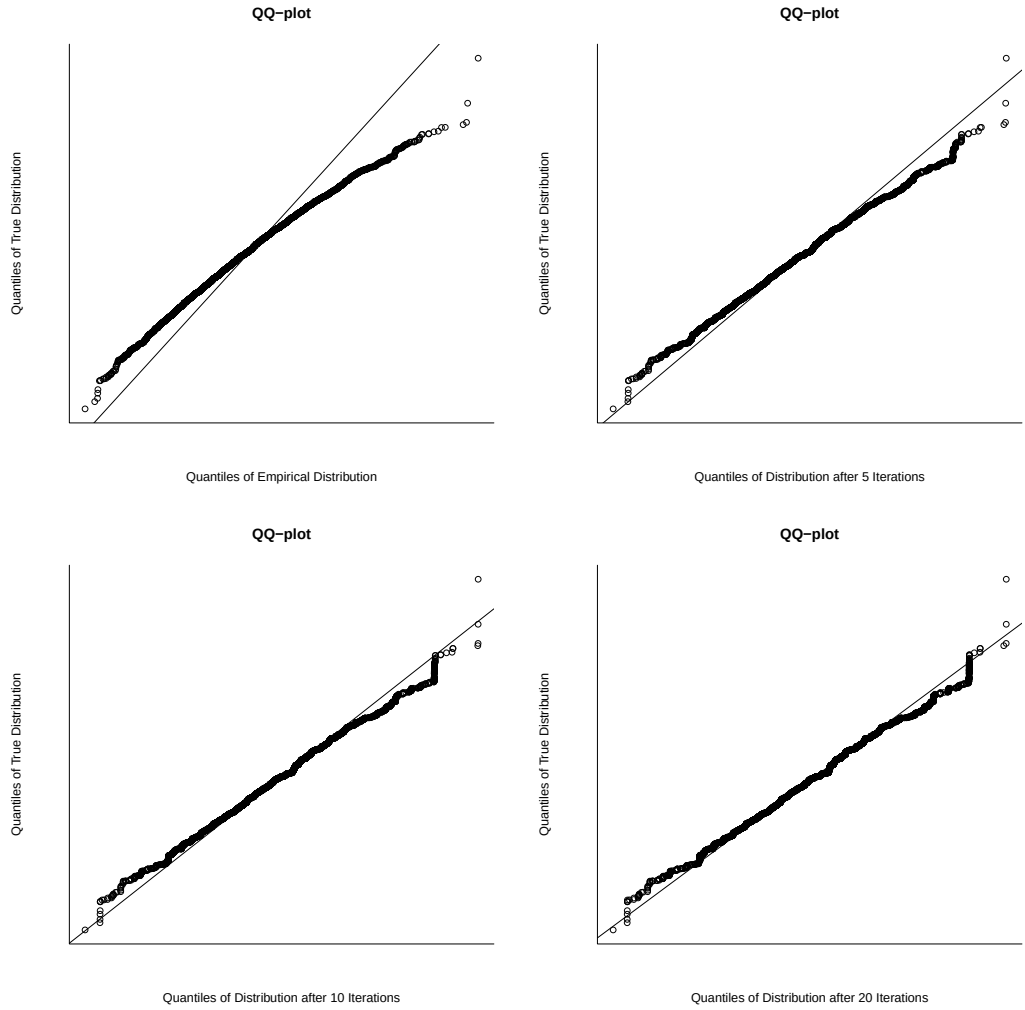

Figure 5.1: Simulation with $m = 20$ and χ^2_{20}. QQ plot for the true variances against the estimated variances. The top left one is about the sample variances. The top right one is about the stochastic approximation with 5 iterations. The bottom left one is the stochastic approximation with 10 iterations and the bottom right one with 20 iterations.

5.5.2 Verification of the Assumption I

In Figure 5.4, we verify our assumption I (5.8):

$$s^n(\alpha)|q^n(\alpha) \sim (q^n(\alpha), \frac{1}{m}).$$

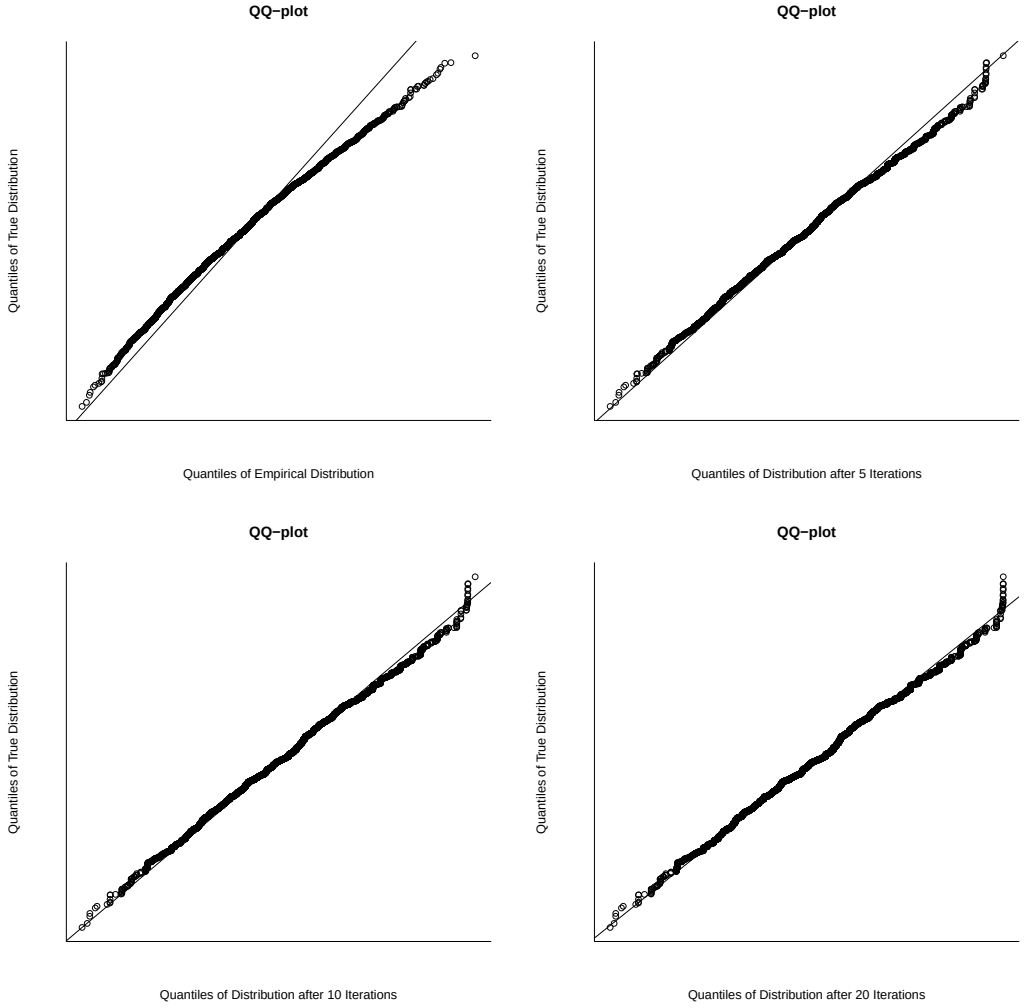

Figure 5.2: Simulation with $m = 20$ and χ^2_{10}. QQ plot for the true variances against the estimated variances. The top left one is about the sample variances. The top right one is about the stochastic approximation with 5 iterations. The bottom left one is the stochastic approximation with 10 iterations and the bottom right one with 20 iterations.

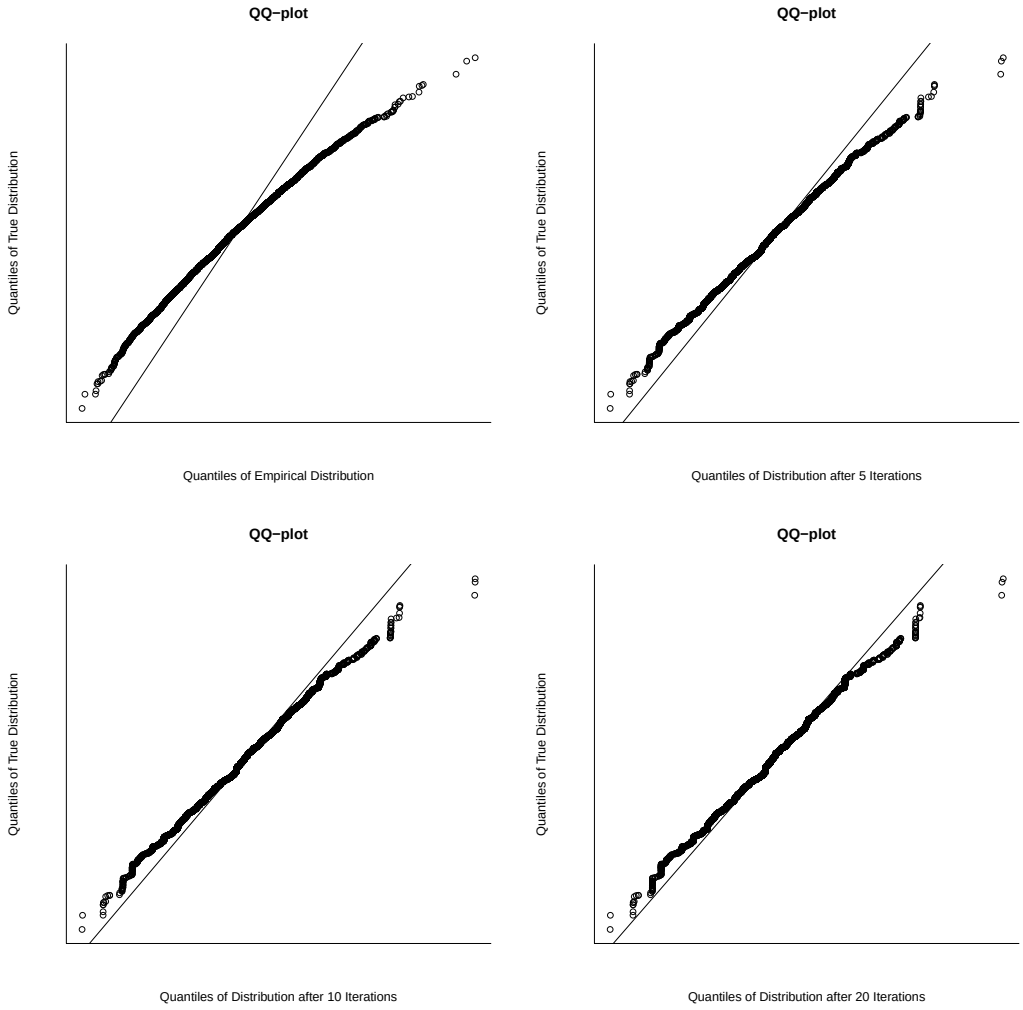

Figure 5.3: Simulation with $m = 20$ and χ^2_{30}. QQ plot for the true variances against the estimated variances. The top left one is about the sample variances. The top right one is about the stochastic approximation with 5 iterations. The bottom left one is the stochastic approximation with 10 iterations and the bottom right one with 20 iterations.

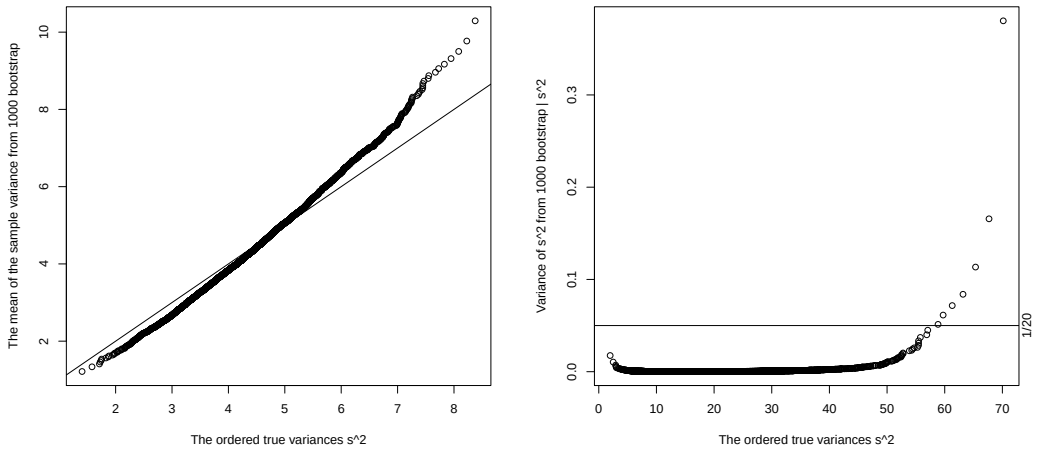

Figure 5.4: Simulation with $m = 20$ and χ^2_{20}.

and assumption II (5.9):

$$s^n(\alpha)|q^n(\alpha) \sim (q^n(\alpha), \frac{(q^n(\alpha))^2}{m}).$$

We sample 5000 variances from χ^2_{20}. For each variance $\sigma^2_g(g = 1, \cdots, 5000)$, we simulate 20 sample points from Normal$(0, \sigma^2_g)$ and get the sample variances $s^2_g(g = 1, \cdots, 5000)$. Denote the ordered sample variances as $s^2_{(g)}(g = 1, \cdots, 5000)$. For each sample variances $s^2_{(g)}$, we simulate 1000 sets of samples with sample size 20 from Normal$(0, s^2_{(g)})$ and get the ordered sample variances $s^2_{(g),b}$ $(b = 1, 2, \cdots, 1000)$. For each g, we get the mean $\overline{s}_{(g)}$ and variances $s^2_{(g),B}$ of $\{s^2_{(g),b} : b = 1, 2, \cdots, 1000\}$ and plot $\overline{s}^2_{(g)}$ (the left one)and $s^2_{(g),B}$ (the right one) against the $s^2_{(g)}$ in Figure 5.4. From the left plot of Figure 5.4, we see our assumption

$$E(s^n(\alpha)|q^n(\alpha)) = q^n(\alpha)$$

is reasonable. The right plot of Figure 5.4 shows that

$$\text{Var}(s^n(\alpha)|q^n(\alpha)) < \frac{1}{m},$$

which seems to suggest that our assumptions about the variances are too much for cases like this.

5.6 Extension to Correlation Matrix Estimation

As we all know, genes do not work independently. They work in groups and a set of genes may impact the same pathways. If we can get the correlations among genes, the analysis will be improved greatly. It is well known that when the sample size is small, compared with the dimension of the measurement space, the correlation matrix estimates become highly variable. Pearson's correlation coefficient is the most popular correlation estimator used in multivariate analysis. However, its performance is poor when applied to microarray data because of the small sample size. The stochastic approximation procedure we talk about in those sections before in this chapter can be applied to estimate correlation matrix for small-sample-size problems. If we vectoralize the correlation matrix, the stochastic approximation can be applied directly. Of course, we need to make sure the approximated matrix is positive definite. In this section, we will talk about improving the correlation matrix estimation by stochastic approximation.

5.6.1 Fisher Transformation

Hypotheses about the value of the population correlation coefficient ρ between variables X and Y can be tested using the Fisher z-transformation applied to the sample correlation coefficient r. The transformation is defined by

$$z = \frac{1}{2} \ln \left(\frac{1+r}{1-r} \right),$$

where ln is the natural logarithm function. If the samples are independently from the same bivariate normal distribution, then we have

$$z \sim \text{Normal}(\frac{1}{2} \ln \left(\frac{1+\rho}{1-\rho} \right), \frac{1}{n-3}).$$

Then we can see that Fisher transformation is an approximate variance-stabilizing transformation for r under the assumption of bivariate normal distribution. This means that the variance of z is approximately constant for all values of the population correlation coefficient ρ. Without the Fisher transformation, the variance of r grows smaller as $|\rho|$ gets closer to 1. Fisher z-transformation guarantees that our assumptions made in the proof will be satisfied.

The procedure introduced before will be applied on sorted Fisher-z transformed sample correlations. Sorting is very important. Without sorting, the improvement is little. To make sure the matrix will be positive definite, we change the negative eigenvalues to a small positive value, say 10^{-3}.

One simulation result is plotted in Figure 5.5, 5.6, and 5.7. It shows clearly that stochastic approximation improves empirical correlation matrix.

5.7 Discussion

In this chapter, we proved the algorithm used in conditional t converges under two assumptions. Sorting is applied in the algorithm to make sure the monotonicity of estimated quantiles. It would be much better if convergence could be shown for the vector of the quantiles, instead of for each quantile. Future work will focus on this part. It is also worth to point out that sorting is very important in the proposed algorithm. In approximating the correlation matrix, theoretically, sorting is not quite necessary as there is no mandatory monotonicity for the correlation coefficients. However, by simulation, we found that without the sorting procedure, stochastic approximation alone cannot make any improvement. We will study this in detail in future.

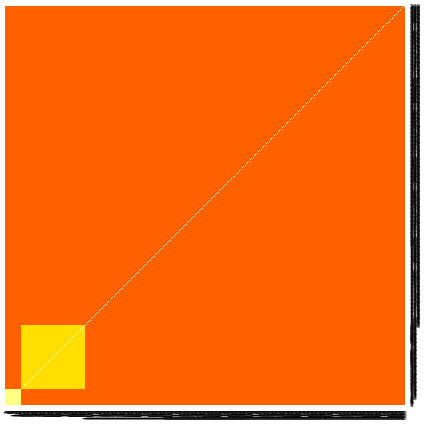

Figure 5.5: Heatmap of the True correlation matrix

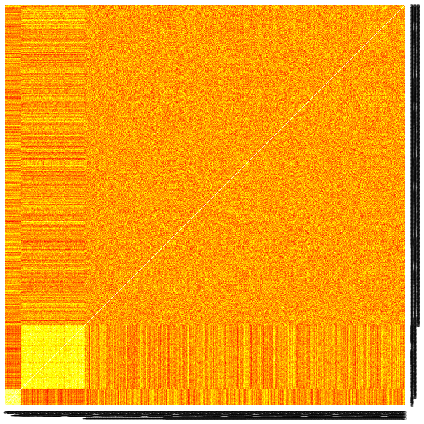

Figure 5.6: Heatmap of the Empirical correlation matrix

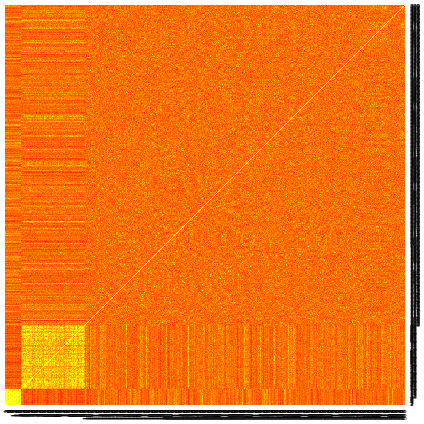

Figure 5.7: Heatmap of the Stochastic Approximated correlation matrix

Chapter 6

Data Analysis about one Novel Experiment

In this chapter, we discuss new development in experiment applying microarray technology. It is believed that when people get cancer, some genes which have effects on the cell growth rate behave wildly. With development of biological technology, the idea of shutting down those wild genes is quite appealing. With new technology, scientists now can introduce additional sequences into the genome. When one gene containing the inserted sequences expresses, the resulting mRNA which is single stranded normally will contain a small part of double-stranded sequence which is quite chemically inert. When the cell tries to interpret the codons of the mRNA and produce the corresponding amino acid chain according to the sequence of codons, this double-stranded sequence interrupts. The production of amino acid chain stops as the cell can not interpret the double-stranded sequence chemically. In this way, the gene expression is interrupted and we call the gene is shut down/silenced/knocked down. Then one question is which genes (biomarkers) should be shut down, i.e., shutting down which genes could reduce the cell growth rate. Along with the method of silencing some gene, some compound is also applied to the cell in purpose of slowing down the cell growth rate. Then the other question is what will the interaction effect between shutting down one gene and the compound be. Scientists have designed some novel experiment addressing these two questions.

In this chapter, we will discuss the data analysis about such experiment. We first give a short introduction about the new technology, then discuss the detailed information about the experiment and about the data we get from the experiment. We then address different issues in the analysis. Finally we show the results of our analysis.

6.1 Introduction of the New Technology

In all organisms, there are two major steps separating a protein-coding gene from its protein: transcription and translation. In transcription step, the gene information must be transcribed from DNA to messenger RNA (mRNA). In translation step, the information is translated from mRNA into protein. A simplified graph of these two steps (transcription and translation) is in Figure 6.1.

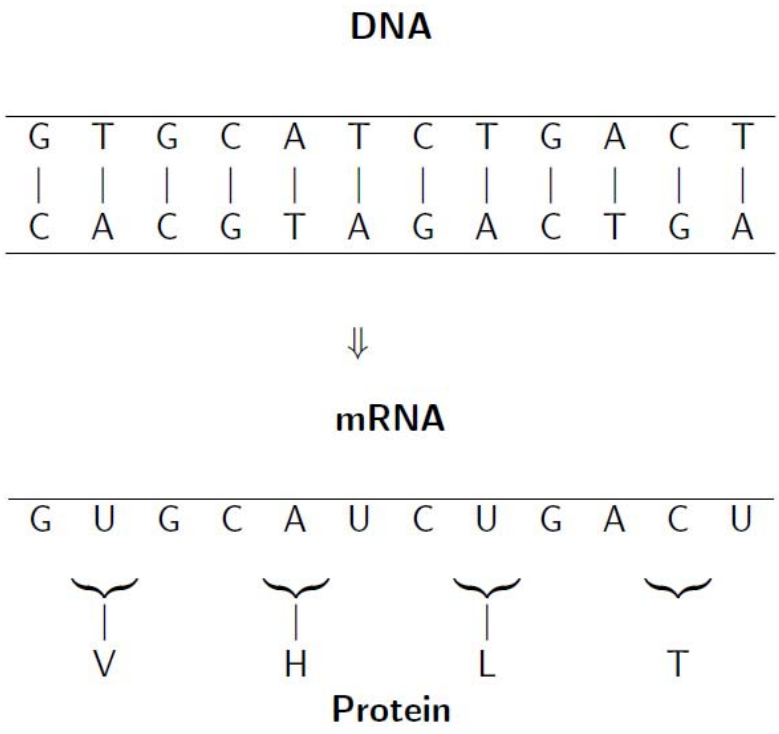

Figure 6.1: Simplified graphical representation of the transcription and translation steps in gene expression.

When genetically engineered molecules are inserted into the host genome, they are transcribed in the nucleus by some polymerase into mRNA. The corresponding mRNA will contain a small double-stranded sequence, which will interrupt the translation of the mRNA: the construction of the amino acid chain is halted, no protein will be produced, therefore, the corresponding gene is silenced. A simplified graph (Figure 6.2) shows a rough idea how the translation step in gene expression process is impacted. Technology development enables scientists to silence/shut down one specific gene by one specific molecule. One gene could have several different molecules which can shut down it: different molecules integrate

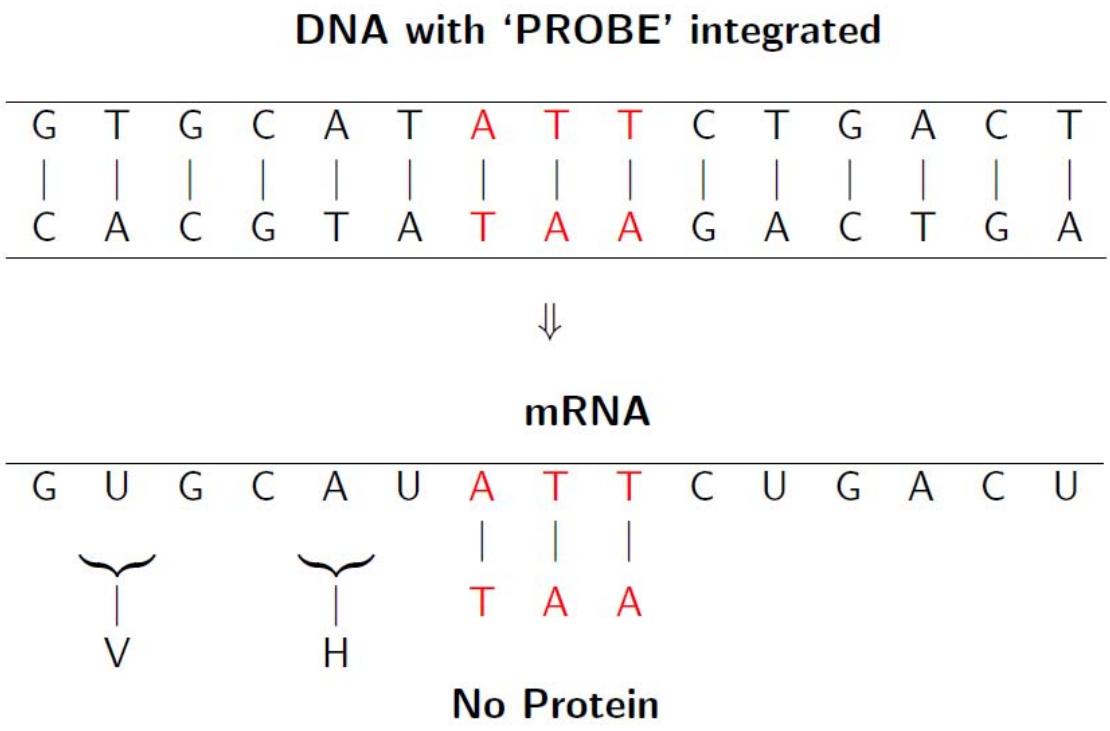

Figure 6.2: Graphical representation of 'PROBE' shutting down one gene

into different places of the gene. However, the efficiency of the shutdown is usually unknown. The molecules are inheritable: when the genome of one cell gets such molecule, the genome of its daughter cells will also contain that molecule. There are large libraries of such molecules targeting thousands of genes in parallel. For convenience, from now on, we will call the molecule 'PROBE'. It is in capital letters because it is different from probe for one gene. However, as one probe represents one gene and one gene can have several probes, we think the name 'PROBE' is convenient for interpretation.

6.2 Our Experiment

The experiment here is about one compound, which can slow down the cell growth rate when added to a collection of cells. The objectives of the experiment are: (1) the effect of knocking down one gene by 'PROBE' on the cell growth rate; (2) the interaction effect between each gene and the compound.

For objective (1), no compound is added. When one gene is knocked down by some

'PROBE', and the cell growth rate is larger than some criterion, we say knocking down this gene will promote the cell growth rate (Denoted as 'GP' - growth promoting); similarly, if the cell growth rate is smaller than the criterion, we say knocking down the gene will inhibit the cell growth rate (Denoted as 'GI' - growth inhibiting); finally, if the cell growth rate does not change when the gene is silenced and no compound is added, we say knocking down the gene has no effect on the cell growth rate (Denoted as 'GN' - growth neutral).

For objective (2), different doses of the compound were added into the cells. As the compound will slow down the cell growth rate, there is a maximum dose level of the compound that can be added into the cells. Any dose level larger than the maximum dose level would kill all the cells. We call the maximum dose level 'clinical dose' and denote it as C. When one gene is knocked down by some 'PROBE', we'd like to know whether there is any interaction effect between the knockdown of the gene and the compound in the concentration level $(0, C)$. The cell growth rate in this interval will be a function of the concentration level, we call it 'growth rate curve'. If the growth rate curve is always below some reference curve, we say knocking down the gene will strengthen the compound effect (denoted as 'S'); if the cell growth rate curve is always above the reference curve, we say knocking down the gene will weaken the compound effect (denoted as 'W'); if the cell growth rate curve is always 'close' to the reference curve, we say knocking down the gene has no interaction with the compound (denoted as 'NI'). However, there are cases when the growth rate curve and the reference curve cross each other in the interval $(0, C)$. Then the interaction effect could be strengthening in one part and weakening in the other, and vice versa. We will not study such cases in detail, instead we say that the interaction effect between knocking down of the gene and the compound is changing (denoted as 'C').

In all, each gene may fall into one of the 12 possible classes listed in Table 6.1. We are most interested in those genes which may interact with the compound, i.e. genes fall into classes 'GN_S', 'GP_S', 'GI_S', 'GN_W', 'GP_W', and 'GI_W', as those genes could be the possible biomarkers that the scientists are after.

When doing the experiment, one pooled 'PROBE' plasmid library packaged in virus was poured into a large collection of cells. Sufficient time was given to make sure almost all 'PROBE's were integrated into some cell. We assume each cell contains at most 1

class	Absence of Compound	Presence of Compound	Abbreviation
1	Growth Neutral	No Interaction	GN_NI
2	Growth Promoting	No Interaction	GP_NI
3	Growth Inhibiting	No Interaction	GI_NI
4	Growth Neutral	Strengthening	GN_S
5	Growth Promoting	Strengthening	GP_S
6	Growth Inhibiting	Strengthening	GI_S
7	Growth Neutral	Weakening	GN_W
8	Growth Promoting	Weakening	GP_W
9	Growth Inhibiting	Weakening	GI_W
10	Growth Neutral	Changing	GN_C
11	Growth Promoting	Changing	GP_C
12	Growth Inhibiting	Changing	GI_C

Table 6.1: 12 possible categorizations describing the effect of knocking down one gene on the cell growth rate and the interaction effect between knocking down one gene and the compound.

'PROBE'. Then cells without 'PROBE' will be removed by puromycine, the remaining cells will be used in our experiment. The aforementioned procedure was repeated to get enough samples. Then different concentration (or dose) levels of the compound will be added to each sample. Let the cells in each sample grow for a certain amount of time. Then using microarray technology, the abundance of cells that host a specific 'PROBE' can be read out as intensities on a probe-specific arbitrary scale. Comparison of intensities at different time points enables estimation of the (exponential) growth rate of cells hosting any given 'PROBE', and hence the effect of knocking down of the associated target gene on cell proliferation.

6.2.1 Data from the Experiment

The Total Number of 'PROBE's and its Distribution

Our experiment involves 54020 'PROBE's (We denote the total number of 'PROBE's as S later for convenience), corresponding to more than 11248 genes (the corresponding genes of some 'PROBE's are missing). Table 6.2 contains the number of genes with different number of 'PROBE's, since one 'PROBE' will shut down one gene and different 'PROBE's could shut down the same gene. From this table, we see there are 9 genes which have only 1 corresponding 'PROBE' that could shut down each of them; most of these genes have 4 or 5

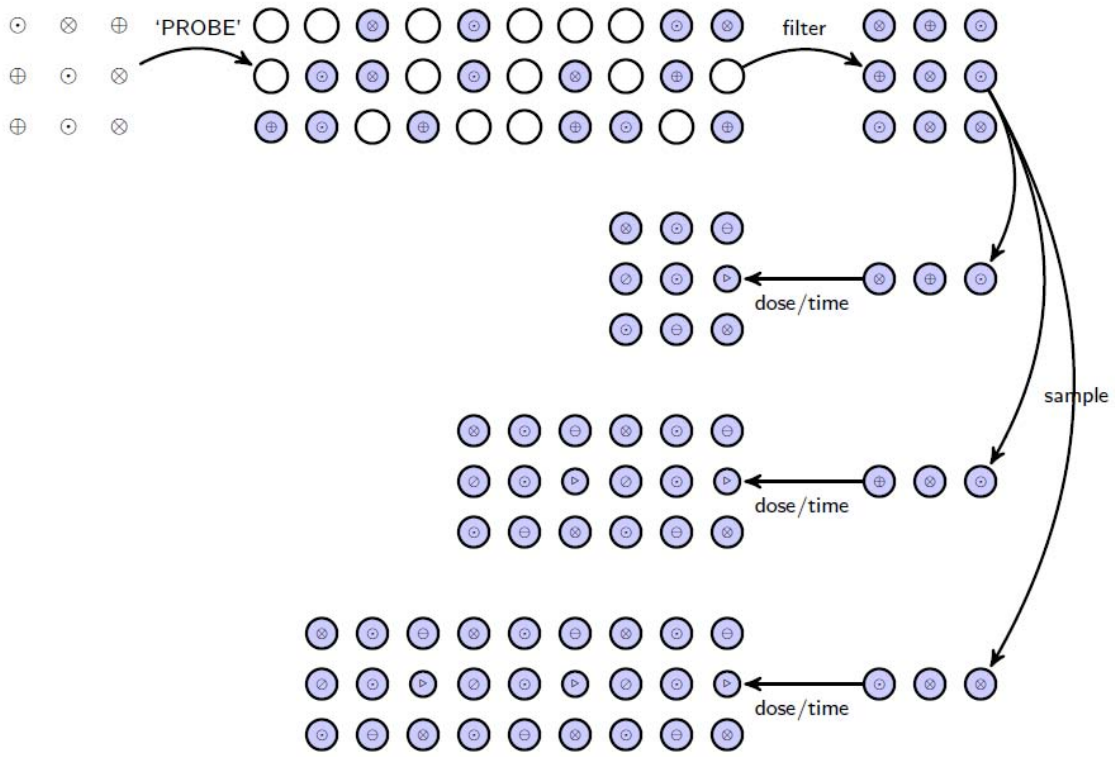

Figure 6.3: Rough idea about the experiment.

'PROBE's: 2220 genes have 4 'PROBE's and 8387 genes have 5 'PROBE's. In the extreme case, there are 31 'PROBE's which will shut down the same gene.

# 'PROBE'	1	2	3	4	5	6	7	8	9
# gene	9	69	406	2220	8387	33	44	22	12
# 'PROBE'	10	11	12	13	14	15	17	31	
# gene	20	11	7	3	2	1	1	1	

Table 6.2: The number of genes with different number of 'PROBE's

Time Points and Concentration Levels

For each concentration level of the compound, data will be collected at three time points: each time point corresponds to one doubling time roughly, which is the period of time required for the number of cells to double. T1 will denote the first doubling time, T2 the second doubling time and T3 the third doubling time. T1, T2, T3 denote different number of days for different concentration level of the compound. As our compound will decrease the growth rate, the larger the concentration level, the longer the days for doubling (as shown

in the right table of Table 6.3). There are 4 concentration levels in our experiment: 0 (no compound), 1, 2.5, 5. Hence, we have 12 experimental conditions (12 combinations of time points and concentration levels of the compound) (see Table 6.3). At each condition, we have 4 samples. Whether these 4 samples are technical replicates and biological replicates is not clear to us. But we assume these samples are independent with each other.

	TimePoint			
Concentration	T1	T2	T3	subTotal
0	4	4	4	12
1	4	4	4	12
2.5	4	4	4	12
5	4	4	4	12
subTotal	16	16	16	48

	Concentration			
days	0	1	2.5	5
7	4	4	4	0
11	4	4	4	4
18	4	4	0	0
20	0	0	4	4
31	0	0	0	4

Table 6.3: Number of samples in each combination of timepoint and concentration

6.2.2 Data We Have

From the experiment, we get the following data:

1. $\log_2$ intensity value $\log_2 I_{k,i}(t,c)$ of the kth ($k = 1, 2, \cdots, S$) 'PROBE' for the ith ($i = 1, 2, 3, 4$) sample at time t ($t = 1, 2, 3$) with compound dose c ($c = 0, 1, 2.5, 5$). By microarray technology, the number of cells with different 'PROBE's could be read and then transformed into $\log_2$ intensity values. The more the cells with one 'PROBE', the larger the intensity values.

2. Total cells counts of the ith sample at time t and compound dose c: $N_{tot,i}(t,c)$ (Table 6.4).

3. For each 'PROBE' modifier, its mapping to the targeted gene.

6.2.3 Date Preprocess

Intensity values from microarray technology could not be compared among different samples. They are the relative intensity of each 'PROBE' in each sample. Moreover, to find out how knocking down the genes and the compound interact with each other, it is preferable to work with the growth rates of cells hosting different 'PROBE's. The cell counts at different

	TimePoint		
Concentration	T1	T2	T3
0	7 days	11 days	18 days
	2×10^9	2.95×10^{10}	7.41×10^{12}
	167654400	3.48×10^{10}	7.78×10^{12}
	2.28×10^9	3.9×10^{10}	6.81×10^{12}
	2.15×10^9	30137554944	6.25×10^{12}
mean	2.03×10^9	3.34×10^{10}	7.06×10^{12}
1	7 days	11 days	18 days
	1.80×10^9	2.70×10^{10}	3.91×10^{12}
	1.40×10^9	2.45×10^{10}	3.17×10^{12}
	1.52×10^9	2.10×10^{10}	3.90×10^{12}
	1.78×10^9	2.74×10^{10}	3.60×10^{12}
mean	1.63×10^9	2.50×10^{10}	3.65×10^{12}
2.5	7 days	11 days	20 days
	1.13×10^9	1.53×10^{10}	2.63×10^{12}
	1.20×10^9	1.42×10^{10}	3.40×10^{12}
	1.20×10^9	1.38×10^{10}	2.92×10^{12}
	1.31×10^9	1.42×10^{10}	2.53×10^{12}
mean	1.21×10^9	1.44×10^{10}	2.87×10^{12}
5	11 days	20 days	31 days
	8.32×10^8	43922781940	4.97×10^{12}
	755155872	4.78×10^{10}	4.60×10^{12}
	814349040	40456974419	5.01×10^{12}
	727993056	35399052552	4.02×10^{12}
mean	7.82×10^8	4.19×10^{10}	4.65×10^{12}

Table 6.4: Total number of cells of sample at each combination

time points with a certain concentration will give us some idea of the cell growth rate. Hence we'd like to get the cell counts for each 'PROBE' in each sample from the intensity values and total cell counts.

The cell counts $N_k(t, c)$ for kth 'PROBE' on the ith sample at time t and concentration c is:

$$N_{k,i}(t, c) = \frac{I_{k,i}(t, c)}{\sum_{k=1}^{S} I_{k,i}(t, c)} \cdot N_{tot,i}(t, c).$$

In Figure 6.4, we plot the cell counts versus time (days) for one 'PROBE', coloring the points by the corresponding concentration of the compound. We see that there is an exponential relationship between the cell counts and time for each concentration, suggesting that log transformation of the cell counts could give linear relationships with the time for each concentration level (the right plot in Figure 6.4). These plots were in agreement with

the general models for cell growth:

$$N_t = N_0 \exp(Kt) \tag{6.1}$$

or

$$\log(N_t) = \log(N_0) + K \cdot t; \tag{6.2}$$

where N_0 is the cell count at the beginning; N_t is the cell count after time t and K is the growth rate:

$$K = \frac{\log(N_t) - \log(N_0)}{t}.$$

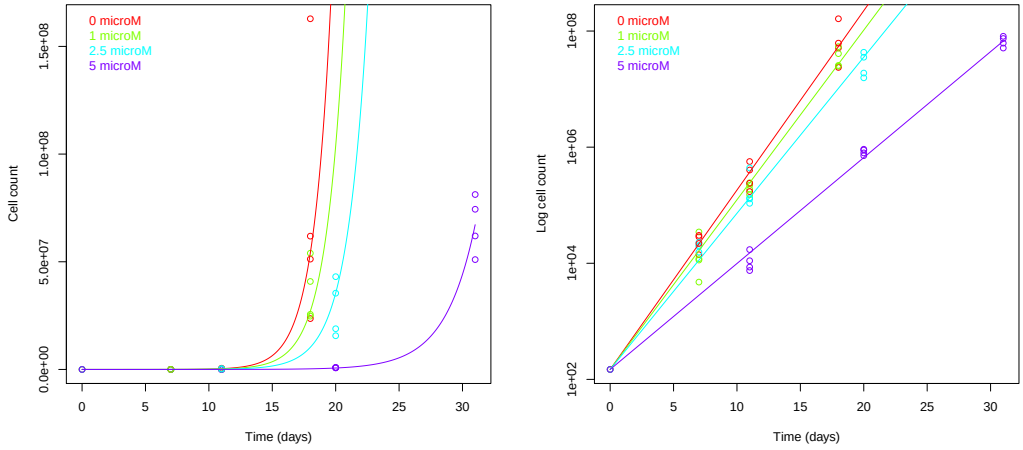

Figure 6.4: The relationship between time and Cell Count for some 'PROBE'

Mean-Variance Relationship

The mean and variance of log cell counts with each 'PROBE' has some pattern as shown in Figure 6.5, which suggests that we should be careful when we apply statistics such as t test which require the mean and variance to be independent.

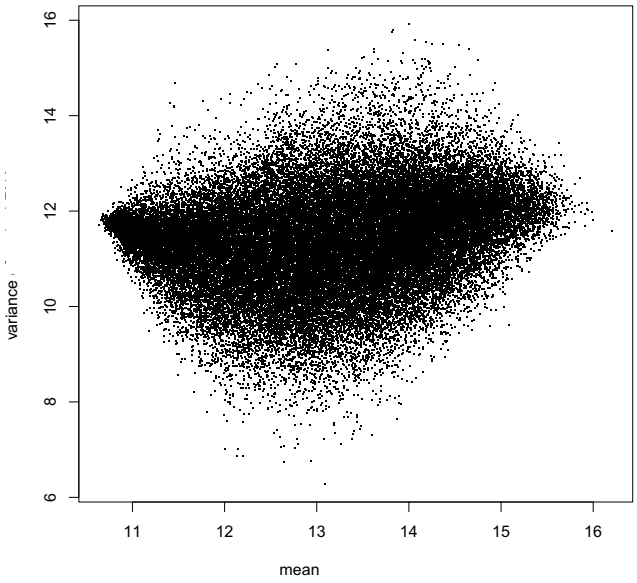

Figure 6.5: The mean and variance of log cell counts for each 'PROBE'

6.3 Modeling for each 'PROBE'

6.3.1 Cell Growth Rate

By assuming that the cell proliferates in a model as (6.1) or (6.2), growth rate of cells with kth 'PROBE' at time t and concentration c can be estimated by:

$$K_k(c) = \frac{\log(N_k(t,c)) - \log(N_0)}{t},$$

where N_0 is the number of cells with kth 'PROBE' at time 0 - the beginning of the experiment. To count the cell numbers, the cells will be killed. Therefore, we do not have the initial count of total cells for each sample. Instead, they will be estimated by grouped linear regression among the different concentrations over time, with common intercept (i.e.

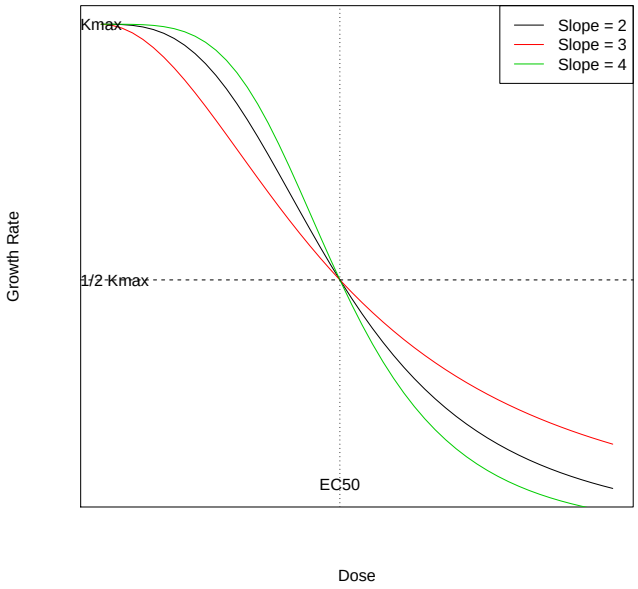

Figure 6.6: Possible dose-response curves

$\log(N_0)$) but concentration specific slopes (Figure 6.4):

$$\log(N_{k,i}(t,c)) = \log(N_0) + I(c=0)\cdot\beta_{k1}t + I(c=1)\cdot\beta_{k2}t + I(c=2.5)\cdot\beta_{k3}t + I(c=5)\cdot\beta_{k4}t.$$

It is easy to see that $K_k(0) = \beta_{k1}, K_k(1) = \beta_{k2}, K_k(2.5) = \beta_{k3}, K_k(5) = \beta_{k4}$.

6.3.2 Dose-Response Model

To study the effect of the compound on the growth rate of cells, dose-response relationship, or exposure-response relationship is ideal. Dose-response model for growth rate $K(c)$ at dose c is assumed as follows:

$$K(c) = \frac{\text{Kmax}}{1 + (c/\,\text{EC50})^{\text{Slope}}},$$

where Kmax is the maximal growth rate, i.e. the cell growth rate without any compound here; EC50 is the concentration level at which 50% of the cell growth is inhibited; Slope is the Hill coefficient. Possible growth rate curves are shown in Figure 6.6, where we can see that Slope will control the shape of the dose-response curve. We also can see that the growth rate curves will cross each other.

6.3.3 Two Possible Models

One intuitive way to build the models for analyzing the data is to concatenate the two models together.

MODEL I: Linear regression plus dose-response model:

Step I: Estimate the growth rate of cell with kth 'PROBE' $K_k(c)$ by grouped linear regression as in section 6.3.1.

Step II: Fit one 3-parameter dose-response model for growth rate (from step I) and concentration c:

$$K_k(c) = \frac{\text{Kmax}}{1 + (c/\,\text{EC50})^{\text{Slope}}}.$$

One problem with model I is the errors in estimating $K_k(c)$ will be ignored in step II. Though working with the growth rates is reasonable, what we can get directly from the data is the cell counts. Instead of estimating the growth rate for further analysis, we suggest working with the cell counts:

MODEL II: Fit one model for log Cell Counts with concentration c and time t:

$$\log(N_k(t,c)) = \beta_0 + t \cdot \frac{\text{Kmax}}{1 + (c/\,\text{EC50})^{\text{Slope}}}$$

where β_0 is the cell counts with kth 'PROBE' at t_0.

By integrating the two steps in model I together, model II accounts for the errors of the two steps in model I, which improves the convergence of the nonlinear part. Building model I for each 'PROBE', we got 216 models out of 54020 models which do not converge; while building model II, we got only 17 models which do not converge. It may seem only a small proportion, but it is definitely some improvement. Therefore, we prefer model II.

From Figure 6.7, we see that there is not much difference in the estimation of the log of cell counts at time 0 using these two models.

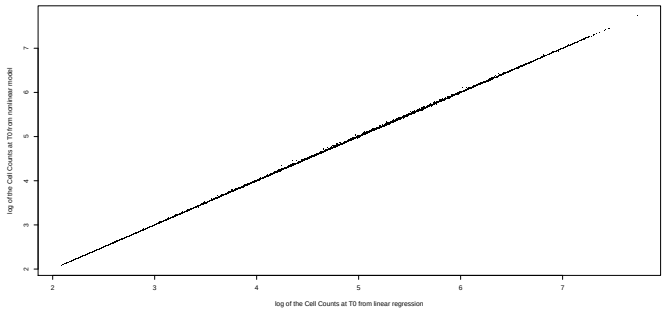

Figure 6.7: Comparing log of cell counts at time 0 from the 2 models

6.4 Define Reference

To comment on the effect of one 'PROBE' on the cell growth rate and the interaction effect between certain 'PROBE'/gene and the compound on the growth rate of cells, we need some benchmark/criterion. Since we do not have any direct observations which could represent the normal growth rate of cells without any effect from the knockdown of any gene and the compound, we should create one benchmark. The implicit assumption of this experiment is that there are only a few genes which will affect cell proliferation and interact with the compound. Therefore, the behavior of the majority of the genes could serve as the benchmark, which we call the "reference".

6.4.1 Method I of Defining the Reference

One way of defining reference is taking the mean of the cell counts for each 'PROBE' on each sample (at time t and with concentration c):

$$N_{r,i}(t,c) = \sum_{k=1}^{S} N_{k,i}(t,c)/S,$$

and treating $\{N_{r,i}(t,c), t = 1,2,3; c = 0,1,2.5,5; i = 1,2,3,4\}$ as the reference 'PROBE'. Then modeling the reference with each 'PROBE'/gene together.

As we can see in Figure 6.8, the variance of the 4 replicates at each time and each dose in the reference is very small. But generally, the variance of the 4 replicates of each 'PROBE' is much larger. When defining the reference this way, it will be inappropriate if we try to model the reference with other 'PROBE' together and assume homoscedasticity at the same time.

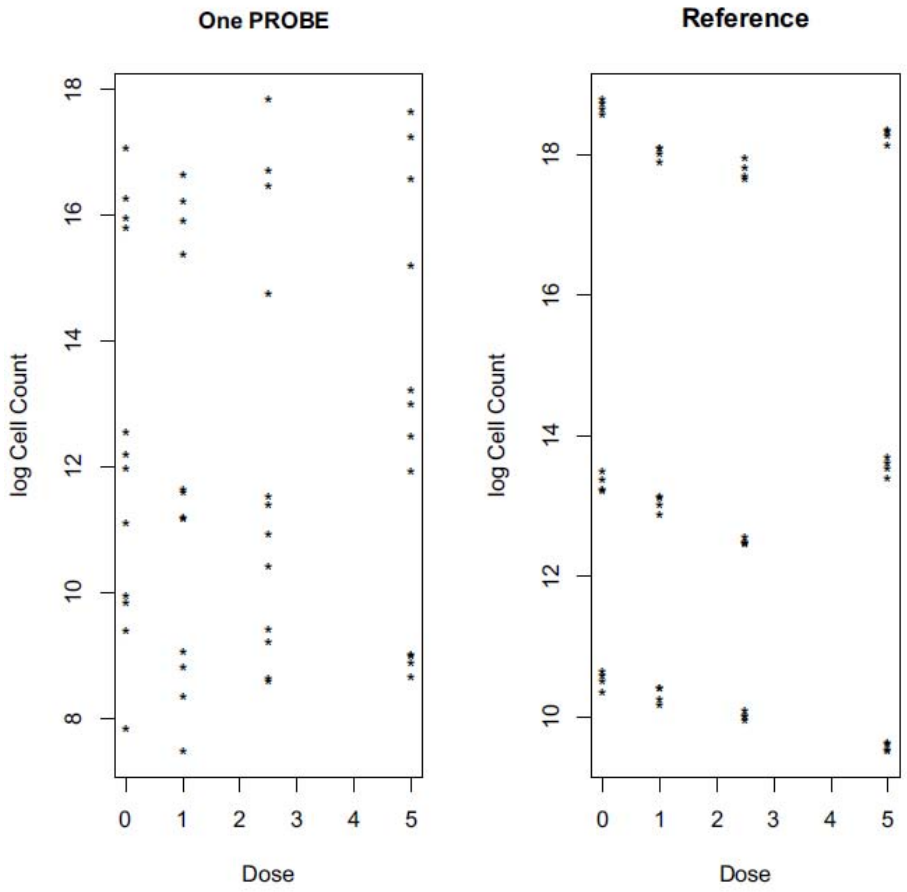

Figure 6.8: Define the mean of each sample as reference

6.4.2 Method II of Defining the Reference

Since our reference is some criterion used in comparison, fixed quantity could be better. We fit our model II to the reference 'PROBE' $N_{r,i}(t, c)$, and define its growth rate curve as the reference. For our data, we get the three parameters of the reference curve as

$$\boldsymbol{\theta}_0 := (\text{Kmax}_0, \text{EC50}_0, \text{Slope}_0) = (0.7290,\ 5.6287,\ 2.2460).$$

6.4.3 Method III of Defining the Reference

Recall that our reference should represent the behavior of the majority of genes. The mean of each sample is a mixture of different behavior and it may not represent the behavior of the majority of genes. Another way of defining reference is: fit the nonlinear model (Model II) for each 'PROBE' separately, and take the spatial median of the parameters as

the reference. In a normed vector space of dimension two or greater, the "spatial median" minimizes the expected L_1 distance

$$\mathbf{a} \mapsto E(\|\mathbf{X} - \mathbf{a}\|)$$

where $\mathbf{X}$ and $\mathbf{a}$ are vectors, if its expectation has a finite minimum. What we get from our data is:

$$\boldsymbol{\theta}_0 := (\text{Kmax}_0, \text{EC50}_0, \text{Slope}_0) = (0.7237,\ 5.5703,\ 2.2749)$$

The marginal means of the 3 parameters are: $(0.7181, 5.5321, 2.2907)$. The marginal medians of the 3 parameters are $(0.7226,\ 5.5600,\ 2.2652)$. Since we have more than 54000 'PROBE's in total, the standard deviations are very small: $(0.0337, 0.3070, 0.3426)$. We prefer this definition since it is more close to the idea of the behavior of the majority of genes.

6.5 Ways of Dealing with Multiple 'PROBE's

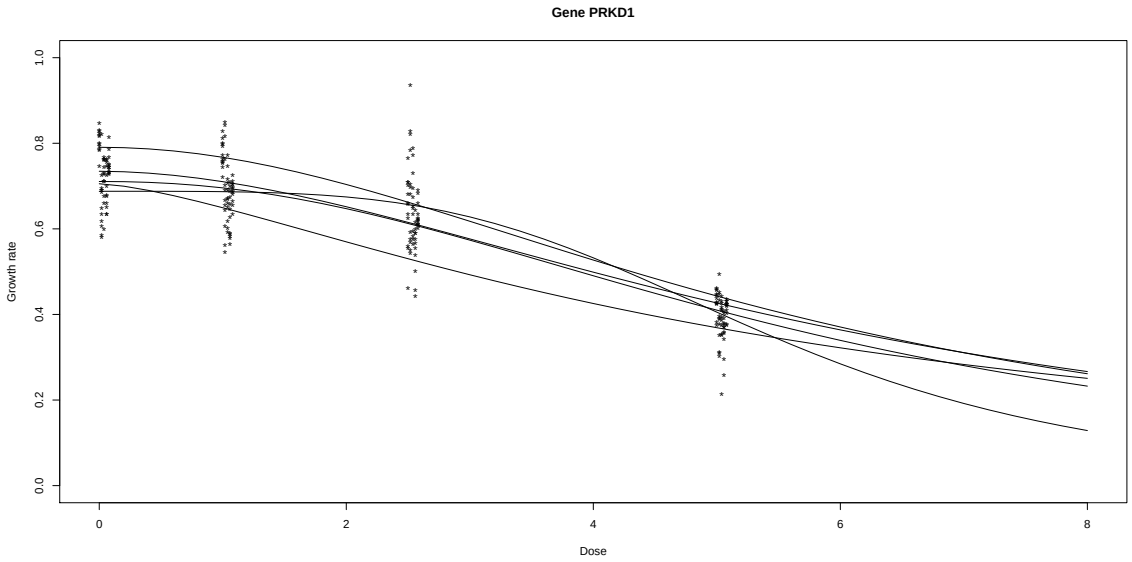

Figure 6.9: 5 'PROBE's for gene 1 with the reference

In Figure 6.9, we plot the growth curve of each 'PROBE' for one gene with the reference curve together. We see that, for genes with multiple 'PROBE's, curves for each 'PROBE' may cross. Directly comparing each 'PROBE' with the reference, we will get different

conclusions for each 'PROBE' of one gene. To get a unified conclusion for the corresponding gene, we need to find a way to analyze all the 'PROBE's to give one reasonable answer for the gene.

One naive way is model all the 'PROBE's for one gene with common growth rate parameters. A more sophisticated way is building mixed effect model for all the 'PROBE's of one gene. Either way, we need to justify the model: when is it reasonable to model all the 'PROBE's with common parameters? Hierarchical clustering is proposed for this purpose. If by some criterion, all the 'PROBE's fall into one cluster, then it is safe to model all the 'PROBE's with common parameters.

6.5.1 Hierarchical Clustering

The hierarchical clustering is based on the following likelihood ratio test: Take 2 'PROBE's of one gene and fit two models for them: one is the full model where each 'PROBE' has its own parameters:

$$\log(N_k(t,c)) = \left(\beta_{01} + t \cdot \frac{\text{Kmax}_1}{1 + (c/\text{EC50}_1)^{\text{Slope}_1}}\right)_{(k=1)} + \left(\beta_{02} + t \cdot \frac{\text{Kmax}_2}{1 + (c/\text{EC50}_2)^{\text{Slope}_2}}\right)_{(k=2)};$$

the other is the simple model where the 2 'PROBE's share one common set of growth rate parameters $(\text{Kmax}, \text{EC50}, \text{Slope})$:

$$\log(N_k(t,c)) = \beta_{01} \cdot (k=1) + \beta_{02} \cdot (k=2) + t \cdot \frac{\text{Kmax}}{1 + (c/\,\text{EC50})^{\text{Slope}}}.$$

The hypothesis test is:

$$H_0 : \text{The Simple Model} \qquad H_1 : \text{The Full Model}$$

If by likelihood ratio test, the simple model is adopted, we group the 2 'PROBE's into 1 cluster.

Suppose we have s 'PROBE's for one gene, we take all the possible C_s^2 groups of 2 'PROBE's and do the likelihood ratio test above. If the largest p value is smaller than the critical value we set up, we stop the process and treat each 'PROBE' as one cluster respectively. Otherwise, we group the 2 'PROBE's with the biggest p value into 1 cluster. Then we treat this new cluster as one new 'PROBE' and continue the clustering process

until either all 'PROBE's are grouped into 1 cluster or the largest p value is smaller than the critical value. The flow of this procedure is in Figure 6.10.

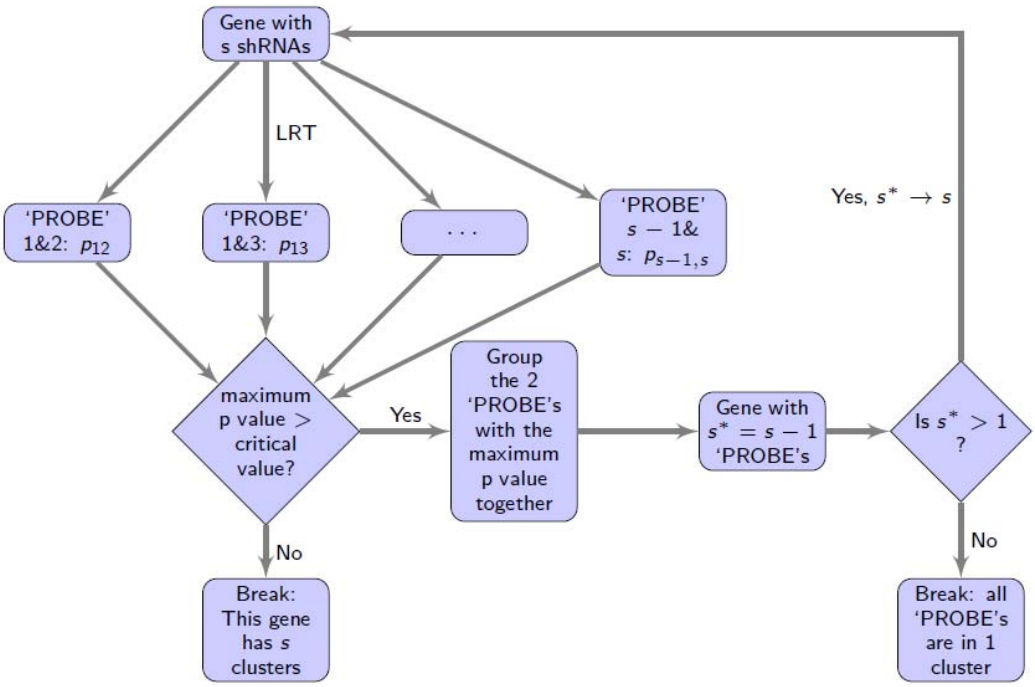

Figure 6.10: Graphical representation of hierarchical clustering

6.6 Model Selection

After clustering the 'PROBE's, each cluster can be modeled in mixed effect model or general nonlinear model dealing with heteroscedasticity.

6.6.1 Mixed Effect Model

Denote the number of cells containing 'PROBE' which belongs to gene i as $N_{ik_i}(t,c)$, where $k_i = 1, \cdots, s$ with s as the total number of 'PROBE's for this gene. Treat the observations for the s 'PROBE' as random sample from one common population.

The model is as follows:

$$\log(N_{ik_i}(t,c))|\mathbf{b}_{k_i} \sim N(\beta_{ik_i} + t \cdot \frac{\text{Kmax}_i + b_{k_i,1}}{1 + (c/(\text{EC50}_i + b_{k_i,2}))^{(\text{Slope}_i + b_{k_i,3})}},\ \sigma^2), \tag{6.3}$$

where

$$\mathbf{b}_{k_i} = (b_{k_i,1},\ b_{k_i,2},\ b_{k_i,3}) \sim N\left(\mathbf{0}, \begin{pmatrix} \sigma_{i1}^2 & 0 & 0 \\ 0 & \sigma_{i2}^2 & 0 \\ 0 & 0 & \sigma_{i3}^2 \end{pmatrix}\right), \quad \forall\ k_i = 1, 2, \cdots, s.$$

6.6.2 General Nonlinear Model

In this model, we assume the log cell counts in each combination of time of each 'PROBE' has a different variance:

$$\log(N_{ik_i}(t,c)) \sim N(\beta_{ik_i} + t \cdot \frac{\text{Kmax}}{1 + (c/\,\text{EC50})^{\text{Slope}}},\ \sigma_{ik_i}^2(t)),$$

where $k_i = 1, 2, \cdots, s$.

6.6.3 Compare these two Models

# 'PROBE'	M:= AIC(MEM)-AIC(GNLS) < -10	$-10 <$ M < 10	M > 10
4	2057 (80.6%)	287 (11.2%)	208 (8.2%)
5	9372 (82.3%)	1287 (11.3%)	727 (6.4%)

Table 6.5: Model selection between mixed effect model and general nonlinear model

From Table 6.5, we see mixed effect model behaves much better than the general nonlinear model, by AIC. Therefore, we will use mixed effect model later.

6.7 Categorize each Cluster of 'PROBE's

The purpose of the experiment is to decide the function of each gene on cell proliferation and its interaction with the compound. In Figure 6.11, we plot several growth curves. The difference showing along the '(1)' arrow is the effect of 'PROBE' irrespective of the compound. The grey dashed line is the reference. We see that cells with the two clusters of 'PROBE's corresponding to the green and light blue curves respectively have smaller growth rates than the reference when there is no compound added. Therefore, such 'PROBE's (i.e. silencing the corresponding genes) are inhibiting the growth rate. On the other hand,

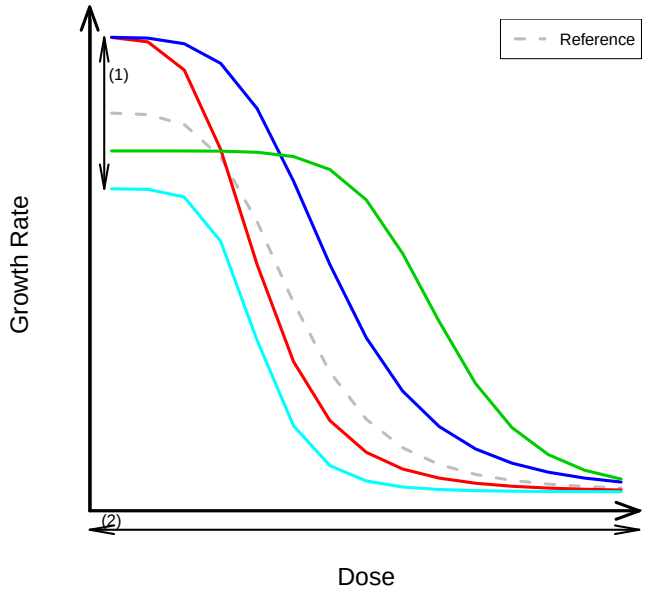

Figure 6.11: Graphical representation of our objective

'PROBE's corresponding to the red and blue curves are growth promoting. As the dose-response model shows, the difference along the '(1)' arrow is accounted by the parameter Kmax.

The difference along '(2)' arrow in Figure 6.11 shows the interaction between each cluster of 'PROBE's and the compound. The interaction effect is difficult to determine from the curves there, since we need to adjust the difference along the '(1)' arrow first. After adjusting the effect on the cell growth rate of each 'PROBE' without the compound, we will have curves as in Figure 6.12. Cases shown in Figure 6.12 are the simplest ones. It will be easy to see that the cluster of 'PROBE's corresponding to the red curve strengthens the compound effect, while the cluster of 'PROBE's corresponding to the blue curve weakens the compound effect.

The idea of determining the effect of one cluster of 'PROBE's without the compound, and the interaction effect of one cluster of 'PROBE's with the compound is shown in Figure 6.13. We compare the Kmax to get the effect of each cluster of 'PROBE's on the cell growth rate; and work with (EC50, Slope) to get the interaction effect.

Suppose from the model in section 6.6, we get the estimates of parameters for the

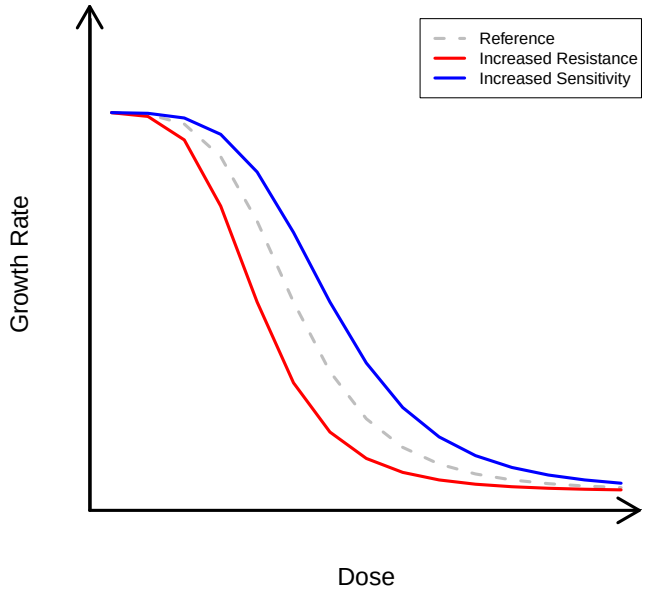

Figure 6.12: Graphical representation of our objective

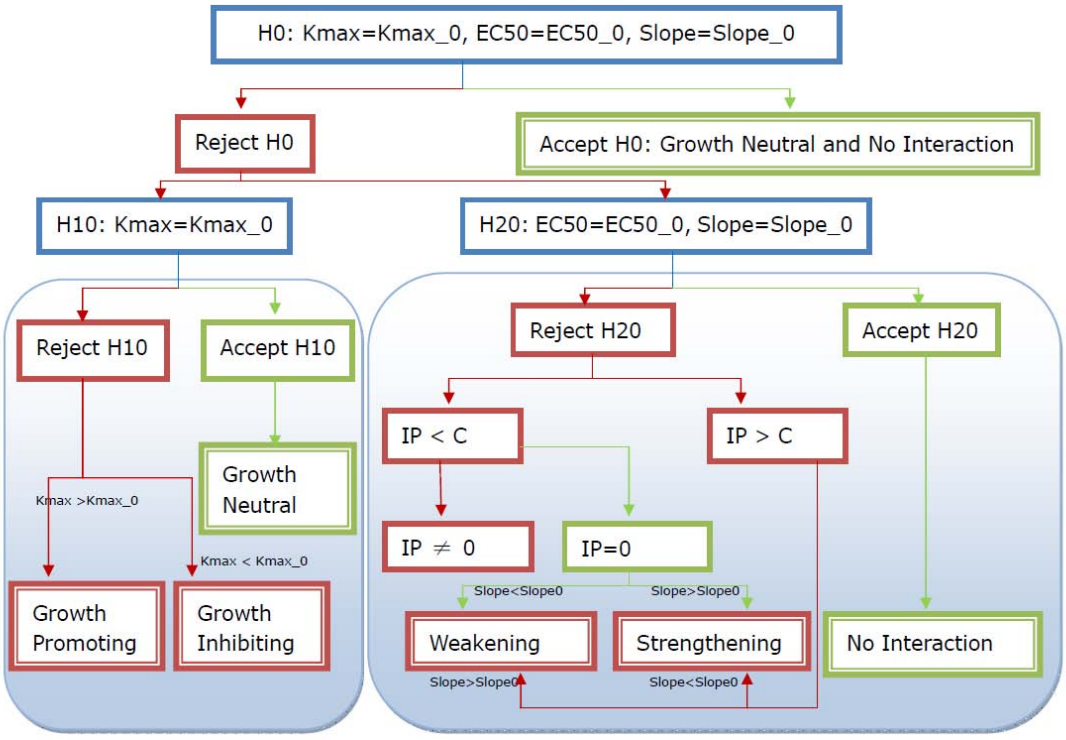

Figure 6.13: Graphical representation of the procedure. IP is the intersection point and C is the clinical dose. The left part is to determine the effect of each 'PROBE'/gene when there is no compound; the right part is to determine the interaction effect between each 'PROBE'/gene with the compound.

dose-response model for each cluster of 'PROBE's: ($\widehat{\text{Kmax}}$, $\widehat{\text{EC50}}$, $\widehat{\text{Slope}}$), along with the estimate of covariance matrix of these three parameters:

$$\hat{\Sigma} = \begin{pmatrix} \hat{\Sigma}_{11} & \hat{\Sigma}_{12} & \hat{\Sigma}_{13} \\ \hat{\Sigma}_{21} & \hat{\Sigma}_{22} & \hat{\Sigma}_{23} \\ \hat{\Sigma}_{31} & \hat{\Sigma}_{32} & \hat{\Sigma}_{33} \end{pmatrix}.$$

As Figure 6.13 shows, before comparing Kmax and (EC50, Slope) separately, we first test the three parameters together by constructing a simultaneous confidence interval for them. A $(1-\alpha)$ simultaneous confidence interval for the 3 parameters is:

$$\{\boldsymbol{\theta} : (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\hat{\Sigma}^{-1}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T < \chi^2_{3,1-\alpha}\} \tag{6.4}$$

If the parameter vector for the reference $\boldsymbol{\theta}_0$ is in the confidence interval, we conclude this cluster behaves similar to the reference, i.e. the major part of all the genes. Then this cluster of 'PROBE's are growth neutral and have no interaction with the compound.

If $\boldsymbol{\theta}_0$ is not in the confidence interval, we'd like to see which one of these three parameters are different from the reference.

6.7.1 Absence of Compound

To determine the effect of the cluster of 'PROBE's on the cell growth rate when there is no compound, we compare its Kmax to that of the reference Kmax_0. If $\text{Kmax} > \text{Kmax}_0$, then the cluster of 'PROBE's is growth promoting; if $\text{Kmax} < \text{Kmax}_0$, the cluster of 'PROBE's is growth inhibiting; if $\text{Kmax} = \text{Kmax}_0$, the cluster of 'PROBE's is growth neutral. Therefore, the problem turns into hypothesis testing: $H_0 : \text{Kmax} = \text{Kmax}_0$ vs $H_1 : \text{Kmax} \neq \text{Kmax}_0$. The following Wald test will be used:

$$\frac{\widehat{\text{Kmax}} - \text{Kmax}_0}{sqrt(\hat{\Sigma}_{11})}. \tag{6.5}$$

If H_0 is accepted, we call this cluster of 'PROBE's are growth neutral. When H_0 is rejected, if $\widehat{\text{Kmax}} > \text{Kmax}_0$, we call this cluster of 'PROBE's are growth promoting; if $\widehat{\text{Kmax}} < \text{Kmax}_0$, we call this cluster of 'PROBE's are growth inhibiting.

6.7.2 Presence of Compound

To determine the interaction of one cluster of 'PROBE' with the compound, we need to compare the adjusted cell growth rate:

$$\frac{1}{1+(c/\mathrm{EC50})^{\mathrm{Slope}}}, \tag{6.6}$$

which adjusts the effect of each cluster of 'PROBE's on the cell growth rate when there is no compound. It is equivalent to determine the following relationship:

$$(\frac{c}{\mathrm{EC50}})^{\mathrm{Slope}} \lessgtr (\frac{c}{\mathrm{EC50_0}})^{\mathrm{Slope_0}}. \tag{6.7}$$

If for all c, we have

$$(\frac{c}{\mathrm{EC50}})^{\mathrm{Slope}} \leq (\frac{c}{\mathrm{EC50_0}})^{\mathrm{Slope_0}},$$

then the cluster of 'PROBE's weakens the compound effect. If for all c, we have

$$(\frac{c}{\mathrm{EC50}})^{\mathrm{Slope}} \geq (\frac{c}{\mathrm{EC50_0}})^{\mathrm{Slope_0}},$$

then the cluster of 'PROBE's strengthens the compound effect. If for all c, we have

$$(\frac{c}{\mathrm{EC50}})^{\mathrm{Slope}} = (\frac{c}{\mathrm{EC50_0}})^{\mathrm{Slope_0}},$$

then the cluster of 'PROBE's has no interaction with the compound. Obviously, the above cases are the simplest situations. In reality, the interaction effect could be changing in the dose/concentration interval we are interested in $(0, C)$, for those cases, we say the interaction effect depending on the concentration. Moreover, the relationship between the parts on the two sides of (6.7) is not so clear with real data.

To determine the interaction effect, we need to apply hypothesis testing on the parameters $(\mathrm{EC50}, \mathrm{Slope})$. First, we test the two parameters together. Construct a $(1-\alpha)$ simultaneous confidence interval for $\boldsymbol{\theta}_1 := (\mathrm{EC50}, \mathrm{Slope})$:

$$\{\boldsymbol{\theta}_1 : (\boldsymbol{\theta}_1 - \hat{\boldsymbol{\theta}}_1)\hat{\Sigma}_1^{-1}(\boldsymbol{\theta}_1 - \hat{\boldsymbol{\theta}}_1)^T < \chi^2_{2,1-\alpha}\} \tag{6.8}$$

where

$$\hat{\Sigma}_1 = \begin{pmatrix} \hat{\Sigma}_{22} & \hat{\Sigma}_{23} \\ \hat{\Sigma}_{32} & \hat{\Sigma}_{33} \end{pmatrix}.$$

If the parameter vector of the reference $\boldsymbol{\theta}_{10} := (\text{EC50}_0, \text{Slope}_0)$ is in the interval, we say this cluster of 'PROBE's have no interaction with the compound. If $\boldsymbol{\theta}_{10}$ is not in the interval, we calculate the intersection point (IP) of the growth rate curves for the cluster of 'PROBE's and the reference:

$$\text{IP} = \left(\frac{\text{EC50}^{\text{Slope}}}{\text{EC50}_0{}^{\text{Slope}_0}} \right)^{\frac{1}{\text{Slope}-\text{Slope}_0}}.$$

If $\text{Slope} > \text{Slope}_0$, we have

$$(\frac{c}{\text{EC50}})^{\text{Slope}} = (\frac{c}{\text{EC50}_0})^{\text{Slope}_0} \cdot (\frac{c}{\text{IP}})^{\text{Slope}-\text{Slope}_0}.$$

When $c < \text{IP}$, we have:

$$(\frac{c}{\text{EC50}})^{\text{Slope}} < (\frac{c}{\text{EC50}_0})^{\text{Slope}_0},$$

then the corresponding adjusted growth rate (6.6) is larger than that of the reference, which means the cluster of 'PROBE's weakens the effect of the drug. When $c > \text{IP}$, we have:

$$(\frac{c}{\text{EC50}})^{\text{Slope}} > (\frac{c}{\text{EC50}_0})^{\text{Slope}_0},$$

then the corresponding adjusted growth rate is smaller than that of the reference, which means the cluster of 'PROBE's strengthens the effect of the drug. The strengthening or weakening effect is shown in Figure 6.14. Therefore, we have seen that the interaction effect is different on the two sides of the intersection point (IP).

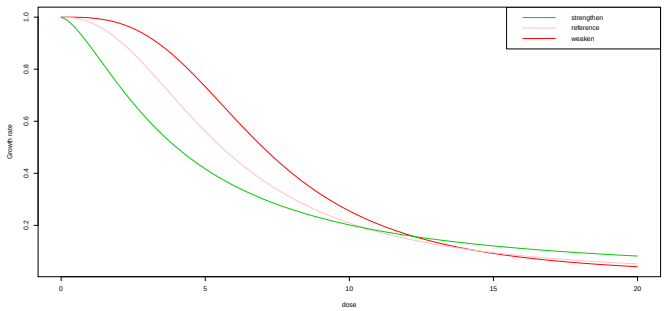

Figure 6.14: Graphical representation of strengthening or weakening effect

Remember, for this experiment, the compound will slow down the cell growth rate; and we are interested in the interaction effect with dose level in the interval $(0, C)$. Hence when

0 <IP< C, the interaction effect between the cluster of 'PROBE's and the compound is changing. However, when IP= 0 or IP$\geq C$, the interaction effect is determined by Slope.

If the intersection point (IP) is larger than the clinical dose (C), when the slope for this cluster is smaller than Slope_0, we say the cluster of 'PROBE's strengthens the effect of the drug; when the slope for this cluster is larger than Slope_0, we say the cluster of 'PROBE's weakens the effect of the drug.

If the intersection point (IP) is smaller than the clinical dose (C), we'd like to see if the intersection point is close to 0, which will be done by hypothesis testing $H_0 : \text{IP} = 0$. Applying delta method to calculate the variance of the intersection point:

$$\begin{aligned}
\text{IP}(\text{EC50}, \text{Slope}) &= (\text{EC50}^{\text{Slope}}/a)^{\frac{1}{\text{Slope}-b}} \\
\frac{\partial \ln \text{IP}}{\partial \text{EC50}} &= \frac{\text{Slope}}{\text{Slope} - b} \cdot \frac{1}{\text{EC50}} \\
\frac{\partial \ln \text{IP}}{\partial \text{Slope}} &= -\frac{b \ln \text{EC50} - \ln a}{(\text{Slope} - a)^2} \\
\text{var}(\text{IP}) &\doteq \left(\frac{\partial \ln \text{IP}}{\partial \text{EC50}} \quad \frac{\partial \ln \text{IP}}{\partial \text{Slope}}\right) \cdot \text{var}((\text{EC50}, \text{Slope})) \cdot \begin{pmatrix} \frac{\partial \ln \text{IP}}{\partial \text{EC50}} \\ \frac{\partial \ln \text{IP}}{\partial \text{Slope}} \end{pmatrix},
\end{aligned}$$

where $a = \text{EC50}_0^{\text{Slope}_0}$ and $b = \text{Slope}_0$. Then we use Wald's statistic to test H_0:

$$\frac{\text{IP}}{\sqrt{\text{var}(\text{IP})}} \sim N(0, 1).$$

If the H_0 hypothesis that 'the intersection point is 0' is retained, when the slope for this cluster is larger than Slope_0, we say the cluster of 'PROBE's strengthens the effect of the drug; when the slope for this cluster is smaller than Slope_0, we say the cluster of 'PROBE's weakens the effect of the drug.

6.8 Analysis of the Data

So far, we have discussed the details of every step in the analysis. The proposed method could be summarized in section 6.8.1.

6.8.1 Method I: HC_MEM_Test

This method is discussed in detail before.

Method I:
- Apply hierarchical clustering to the different 'PROBE's for one gene.
- Fit mixed effect model for each cluster.
- Define reference as in section 6.4.3: spatial median of the parameters for all 'PROBE's.
- To determine the effect of each cluster of 'PROBE's on the cell growth rate with and without the compound, we go through the steps listed in the flow chart in Figure 6.13 as discussed in section 6.7.

6.8.2 Method II: MEM_Test

We would also like to mention another method here: ignore the possibility that each 'PROBE' for one gene may behave differently. Instead, we simply treat each 'PROBE' as some replication of one gene and build mixed effect model for all 'PROBE's of one gene. Comparing the method to the one before, we could justify the hierarchical clustering in Method I.

Method II:
- Fit mixed effect model for all 'PROBE's of each gene;
- Go through the steps listed in the flow chart in Figure 6.13 as discussed in section 6.7.

6.9 Results

6.9.1 Method I: Hierarchical Clustering and Mixed Effect Model

The categorization of genes with different 'PROBE's using Method I (section 6.8.1) is summarized in Table 6.11 (for genes with 1 – 7 'PROBE's), Table 6.6 (for genes with 8 'PROBE's), Table 6.7 (for genes with 9 'PROBE's), Table 6.8 (for genes with 10 'PROBE's), Table 6.9 (for genes with 11 'PROBE's), Table 6.10 (for genes with more than 12 'PROBE's). For genes with $n (3 \leq n \leq 7)$ 'PROBE's, if we get $n-1$ 'PROBE's in one cluster, this cluster of 'PROBE's (cluster I) will represent the gene. The conclusions about the effect of cluster I on the cell growth rate without the compound and the interaction effect between

the compound and cluster I will be the conclusions about shutting down the corresponding gene. However, when we don't get such clusters, for example, if for one gene with 5 'PROBE's, by hierarchical clustering, 3 'PROBE's are grouped into one cluster, and the other 2 'PROBE's are grouped into another cluster, then we are not sure what we can say about the corresponding genes. Therefore, those genes were excluded in Table 6.11. Of course, we can use the cluster containing 3 'PROBE's as the represent of the corresponding gene. But for now, we do not consider those cases. For genes with more than 7 'PROBE's, we showed the detail categorization of each cluster of one gene in those tables. In method I, the critical values used in hierarchical clustering and in hypothesis testing or simultaneous intervals of the flow chart (Figure 6.13: to determine the effect of the cluster of 'PROBE's without the compound and the interaction effect between the cluster of 'PROBE's and the compound) are all 0.001. Obviously, other critical values can be used. But we consider the case with 0.001 here.

By Table 6.11, we see that shutting down most genes are growth neutral and have no interaction with the compound (denote as "GN_NI"): 8 genes out of 9 with only 1 'PROBE' are GN_NI; 64 out of 69 genes with 2 'PROBE's are GN_NI; 301 out of 409 genes with 3 'PROBE's are GN_NI; 1544 out of 2231 genes with 4 'PROBE's are GN_NI; 4964 out of 8373 genes with 5 'PROBE's are GN_NI; 20 out of 33 genes with 6 'PROBE's are GN_NI; and 21 out of 44 genes with 7 'PROBE's are GN_NI. These support the assumption we made when defining the reference: only shutting down a small proportion of genes will impact the cell growth rate and have interaction with the compound.

The columns "GP_NI" and "GI_NI" are about genes shutting down of which may promote or inhibit the cell growth rate but they have no interaction with the compound, i.e., shutting down them will not strengthen or weaken the effect of the compound. In Table 6.11, there are 111 genes in column "GP_NI" and 208 genes in column "GI_NI".

In the highlighted columns of Table 6.11, numbers of genes shutting down of which may strengthen or weaken the effect of the compound are listed. We can see that 23 genes with 3 'PROBE's are "GN_S", i.e., shutting down those 23 genes have no impact on the cell growth rate when there is no compound but will strengthen the effect of the compound; 3 genes with 3 'PROBE's are "GN_W", i.e., shutting down those 3 genes have

no impact on the cell growth rate when there is no compound but will weaken the effect of the compound. Similarly, there are 131 genes with 4 'PROBE's which are GN_S; and 83 genes with 4 'PROBE's are GN_W. 227 genes with 5 'PROBE's are GN_S and 155 genes with 5 'PROBE's are GN_W.

The column "GI_S" is about the number of genes shutting down of which are growth inhibiting when there is no compound and will strengthen the effect of the compound (i.e., when there is no compound, shutting down those genes will inhibit the cell growth rate; moreover, when the compound is added, shutting down those genes will slow down the cell growth rate even more.). The column "GP_W" is about the number of genes shutting down of which are growth promoting when there is no compound and will weaken the effect of the compound. As we have already know the compound will slow down the cell growth rate, shutting down genes in these two columns are consistent with and without the compound. By Table 6.11, we can see there are 28 genes in column "GI_S" and 35 genes in column "GP_W".

The column "GP_S" is about the number of genes shutting down of which are growth promoting when there is no compound but will strengthen the effect of the compound. The column "GI_W" is about the number of genes shutting down of which are growth inhibiting when there is no compound but will weaken the effect of the compound, which is quite the opposite of "GP_S". Shutting down those genes in these two columns will get opposite effects when the compound is added. By Table 6.11, we see that there are 7 genes in column "GP_S" and 5 genes in column "GI_W".

The columns "GN_C", "GP_C" and "GI_C" are about the genes whose interaction effect with the compound will be changing with the concentration level of the compound. The scientists are not sure about what to do with them right now.

When genes with more than 7 'PROBE's, it is difficult to get a unified answer with hierarchical clustering. But as there are not many genes with more than 7 'PROBE's, we show the details of the categorization of each cluster of 'PROBE's in Tables 6.6, 6.7, 6.8, 6.9, and 6.10. We need more information from the scientists to do further analysis with those genes.

		No Compound			With Compound			
# 'PROBE's	genes	Neut	Prom	Inhi	NoInter	Stren	Weaken	Changing
8 (0.001)	8	8	0	0	8	0	0	0
	1	8	0	0	0	0	0	8
	1	8	0	0	7	1	0	0
	1	8	0	0	6	2	0	0
	4	8	0	0	4	4	0	0
	1	4	0	4	4	4	0	0
	1	6	2	0	4	4	0	0
	1	8	0	0	0	5	0	3
	1	8	0	0	0	5	3	0
	1	8	0	0	3	5	0	0
	1	8	0	0	0	4	0	4
	1	3	0	5	5	0	0	3

Table 6.6: Categorization of genes with 8 'PROBE's using method I with critical value 0.001. There are 22 genes in total. We have 8 genes which are growth neutral and have no interaction with the compound (the 1st row). There is 1 gene which is growth neutral and the interaction effect with the compound is changing with the concentration level (the 2nd row). The two genes on the 3rd and 4th rows could be taken as growth neutral and having no interaction with the compound. It is difficult to say anything about other genes.

6.9.2 Method II: Mixed Effect Model

The categorization of genes with different 'PROBE's using Method II (section 6.8.2) is summarized in Table 6.12. Again, we see that most of genes are growth neutral and have no interaction with the compound. By this method, relatively less genes fall into the highlighted columns. More genes were classified as having no interaction with the compound. The critical value used is 0.001.

	No Compound			With Compound			
# 'PROBE's	Neut	Prom	Inhi	NoInter	Stren	Weaken	Changing
9 (0.001)	9	0	0	9	0	0	0
	9	0	0	0	9	0	0
	9	0	0	8	1	0	0
	9	0	0	7	0	0	2
	7	0	2	7	0	2	0
	5	0	4	5	4	0	0
	9	0	0	0	6	0	3
	9	0	0	0	2	0	7
	4	5	0	5	4	0	0
	9	0	0	5	4	0	0
	9	0	0	5	2	2	0
	9	0	0	6	3	0	0

Table 6.7: Categorization of genes with 9 'PROBE's using method I with critical value 0.001. There are 12 genes in total. We have 1 genes which are growth neutral and have no interaction with the compound (the 1st row). There is 1 gene which is growth neutral and will strengthen the effect of the compound (the 2nd row). The third genes on the 3rd, 4th and 5th rows could be taken as growth neutral and having no interaction with the compound. It is difficult to say anything about other genes.

		No Compound			With Compound			
# 'PROBE's	genes	Neut	Prom	Inhi	NoInter	Stren	Weaken	Changing
10 (0.001)	3	10	0	0	10	0	0	0
	2	10	0	0	8	2	0	0
	1	10	0	0	8	0	0	2
	2	10	0	0	7	3	0	0
	1	10	0	0	0	3	7	0
	1	7	0	3	10	0	0	0
	1	10	0	0	6	4	0	0
	1	10	0	0	4	6	0	0
	1	0	5	5	10	0	0	0
	1	7	3	0	7	0	3	0
	1	7	3	0	7	0	0	3
	1	1	3	6	10	0	0	0
	1	10	0	0	0	6	4	0
	1	6	0	4	10	0	0	0
	1	10	0	0	6	4	0	0
	1	4	0	6	6	4	0	0

Table 6.8: Categorization of genes with 10 'PROBE's using method I with critical value 0.001. There are 12 genes in total. We have 3 genes which are growth neutral and have no interaction with the compound (the 1st row). The genes on the 2nd, 3rd, 4th and 6th rows could be taken as growth neutral and having no interaction with the compound. The gene on the 5th row could be taken as growth neutral and weakening the effect of the compound. It is difficult to say anything about other genes.

	No Compound			With Compound			
# 'PROBE's	Neut	Prom	Inhi	NoInter	Stren	Weaken	Changing
11 (0.001)	11	0	0	11	0	0	0
	11	0	0	8	3	0	0
	11	0	0	3	8	0	0
	8	0	3	8	3	0	0
	4	7	0	9	2	0	0
	7	4	0	7	4	0	0
	11	0	0	6	3	2	0
	4	0	7	11	0	0	0
	4	0	7	7	0	4	0
	11	0	0	4	0	7	0
	9	0	2	6	0	2	3

Table 6.9: Categorization of genes with 11 'PROBE's using method I with critical value 0.001. There are 11 genes in total. The gene on the 1st row is growth neutral and having no interaction with the compound. The genes on the 2nd and 4th rows could be taken as growth neutral and having no interaction with the compound. The gene on the 3rd row could be taken as growth neutral and strengthening the interaction effect of the compound. It is difficult to say anything about other genes.

	No Compound			With Compound			
# 'PROBE's	Neut	Prom	Inhi	NoInter	Stren	Weaken	Changing
12 (0.001)	12	0	0	12	0	0	0
	12	0	0	8	4	0	0
	12	0	0	9	0	0	3
	10	2	0	2	3	0	7
	10	0	2	2	7	3	0
	9	0	3	12	0	0	0
	7	0	5	12	0	0	0
13 (0.001)	13	0	0	8	5	0	0
	10	0	3	7	6	0	0
	8	5	0	4	1	8	0
14 (0.001)	11	0	3	9	5	0	0
	10	4	0	9	0	5	0
15 (0.001)	9	6	0	6	9	0	0
17 (0.001)	8	4	5	13	0	0	4
31 (0.001)	21	4	6	5	20	0	6

Table 6.10: Categorization of genes with more than 12 'PROBE's using method I with critical value 0.001. It is difficult to give a unified answer for one gene with more than 12 'PROBE's. We need more information from the scientists about these situations.

#'PROBE'	#Gene	GN_NI	GP_NI	GI_NI	**GN_S**	**GP_S**	**GI_S**	**GN_W**	**GP_W**	**GI_W**	GN_C	GP_C	GI_C
1(0.001)	9	8	0	0	0	0	0	0	0	0	1	0	0
2 (0.001)	69	64	1	0	0	0	0	0	0	0	1	0	0
3 (0.001)	409	301	4	3	23	0	3	3	1	0	5	1	0
4 (0.001)	2231	1544	32	48	131	5	6	83	14	1	36	5	0
5 (0.001)	8373	4964	74	162	227	2	19	155	20	4	72	8	3
6 (0.001)	33	20	0	0	1	0	0	0	0	0	0	0	0
7 (0.001)	44	21	1	0	2	0	0	3	0	0	0	0	0

Table 6.11: Categorization of genes with 1-7 'PROBE's using method I. The table is about those genes whose 'PROBE's (suppose n 'PROBE's ($3 \leq n \leq 7$)) are grouped into 2 clusters by hierarchical clustering, where one cluster contains only one 'PROBE'. The other cluster, with $n-1$ 'PROBE's, will represent the corresponding gene. For genes with 2 'PROBE's, we only include the cases where the 2 'PROBE's are grouped into one cluster by hierarchical clustering.

'PROBE'	GN_NI	GP_NI	GI_NI	**GN_S**	**GP_S**	**GI_S**	**GN_W**	**GP_W**	**GI_W**	GN_C	GP_C	GI_C
2 (0.001)	65	1	0	1	0	0	1	0	0	1	0	0
3 (0.001)	360	5	6	24	0	3	3	1	0	6	1	0
4 (0.001)	1981	19	38	101	5	3	44	5	2	30	2	1
5 (0.001)	7778	56	204	153	1	10	92	4	1	67	4	2
6 (0.001)	32	0	1	0	0	0	0	0	0	0	0	0
7 (0.001)	39	0	0	3	0	0	1	0	0	1	0	0
8 (0.001)	19	0	1	1	0	0	0	0	0	1	0	0
9 (0.001)	10	0	1	1	0	0	0	0	0	0	0	0
10 (0.001)	18	0	1	1	0	0	0	0	0	0	0	0
11 (0.001)	9	0	2	0	0	0	0	0	0	0	0	0
12 (0.001)	7	0	0	0	0	0	0	0	0	0	0	0
13 (0.001)	3	0	0	0	0	0	0	0	0	0	0	0
14 (0.001)	1	0	1	0	0	0	0	0	0	0	0	0
15 (0.001)	1	0	0	0	0	0	0	0	0	0	0	0
17 (0.001)	1	0	0	0	0	0	0	0	0	0	0	0
31 (0.001)	0	0	0	1	0	0	0	0	0	0	0	0

Table 6.12: Categorization of genes using model II: building mixed effect model for all 'PROBE's of each gene. 0.001 is the critical value we used.

6.10 Comparing HC_MEM_MD (0.001) and MEM_MD (0.001)

Comparisons the results about the two methods on genes with 5 'PROBE's are shown in Table 6.13. We can safely say that for a large proportion of genes, these two methods give almost the same results.

'PROBE'	NN	PN	IN	**NS**	**PS**	**IS**	**NW**	**PW**	**IW**	NC	PC	IC
5 (I)	4964	74	162	227	2	19	155	20	4	72	8	3
5 (II)	7778	56	204	153	1	10	92	4	1	67	4	2
5 (common)	4933	33	107	127	0	7	57	4	1	45	2	0

Table 6.13: Compare the results about the two methods on genes with 5 'PROBE's, more comparisons can be found in Table 6.11 and 6.12. Here, the columns are named by two characters. The first one indicates the effect of knocking down the gene on the cell growth rate when there is no compound: 'N' means 'growth neutral'; 'P' means 'growth promoting'; 'I' means 'growth inhibiting'. The second one indicates the interaction effect between knocking down the gene and the compound: 'N' means 'No Interaction'; 'S' means 'Strengthening the effect of the compound'; 'W' means 'Weakening the effect of the compound' and 'C' means 'the interaction effect is changing with the compound level'. The row '5(I)' is the result of categorizing the genes with method I. The row '5(II)' is the result of method II. The row '5(common)' is the number of genes which are categorized into the same group by the two methods.

In Table 6.13, we see that the numbers of genes in each column do not differ quite much. However, the common genes which are classified into each column by these two methods are not as many as we would hope, which means these two methods would give different answers for one gene. Let's check the columns 'PS' and 'IW' here, since there are not many genes in those two columns.

First, Let's check the column 'PS', where the genes are categorized as 'Growth Promoting and Strengthening the compound effect'. By method I, 2 genes fall into this column. However, we find out that these 2 genes are categorized as 'Growth Neutral and No Interaction with the Compound' by method II. The growth rate curves and the adjusted growth rate curves of these 2 genes are plotted in Figure 6.15. By hierarchical clustering, each of these 2 genes has two clusters: one cluster containing 4 'PROBE's and one cluster containing 1 'PROBE'. Black and red curves are the growth rate curves for each cluster. The purple curve is the growth rate curve by method II, which always lies between the growth rate curves of the two clusters. The gene which is categorized as 'PS' by method II is plotted in Figure 6.16. By hierarchical clustering, those 5 'PROBE's of this gene were

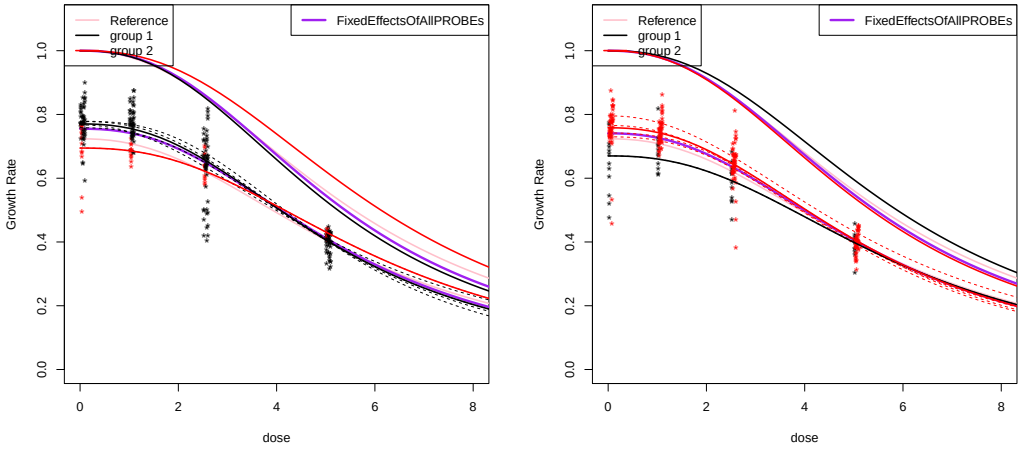

Figure 6.15: The two genes which are categorized as 'Growth Promoting and Strengthening the compound effect' by method I, but are categorized as 'Growth Neutral and No Interaction with the Compound'. Black and red curves are the growth rate curves for each cluster. The purple curve is the growth rate curve by method II. The pink curve is the reference. Growth rate curves with Kmax = 1 are adjusted growth rate curves.

grouped into 2 clusters: one cluster contains 3 'PROBE's and the other cluster contains 2 'PROBE's. The first cluster is categorized as 'Growth Neutral and the interaction effect is Changing'; while the second cluster is categorized as 'Growth Neutral and the interaction effect is Strengthening'.

Next, Let's check the column 'IW', where the genes are categorized as 'Growth Inhibiting and Weakening the compound effect' ('IW'). By method I, 4 genes fall into this column. Among them, 1 gene is also categorized as 'IW' by method II. However, we find out that these other 3 genes are categorized as 'Growth Neutral and No Interaction with the Compound' ('NN') by method II. The growth rate curves and the adjusted growth rate curves of these genes are plotted in Figure 6.17 and 6.18. The plot for the common gene is the left one in Figure 6.17. The 5 'PROBE's are grouped into one cluster by hierarchical clustering. The other three plots in these two Figures are for these three genes which are classified as 'IW' by method I but are categorized as 'NN' by method II. Those 'PROBE's for the three genes are grouped into two clusters respectively: one cluster consists of 4 'PROBE's and the other consists of 1 'PROBE'. The clusters containing 4 'PROBE's are categorized as 'IW'. When applying method II to these three genes, the cluster containing 1 'PROBE' may have

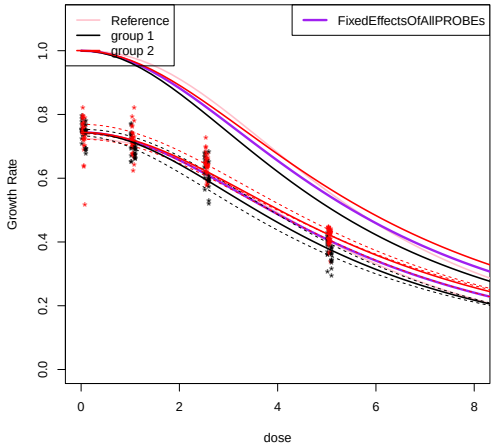

Figure 6.16: The gene which is categorized as 'Growth Promoting and Strengthening the compound effect' by method II, but can not be categorized by method I since the two clusters (one contains 3 'PROBE's and the other contains 2 'PROBE's) give different opinions: the first cluster claims 'Growth Neutral and the interaction effect is Changing'; while the second cluster claims 'Growth Neutral and the interaction effect is Strengthening'. Black and red curves are the growth rate curves for each cluster. The purple curve is the growth rate curve by method II. The pink curve is the reference. Growth rate curves with Kmax = 1 are adjusted growth rate curves.

large leverage and drag the fixed effect of the other 4 'PROBE's towards the reference, as we see in Figure 6.17 and Figure 6.18. In cases like these, we prefer the results given by method I.

6.10.1 Conclusion

The advantage of method II is obvious: we can always get one unified answer for each gene, since hierarchical clustering may split the 'PROBE's into two or more clusters and each cluster may give a different answer. But in cases like the gene in the right plot of Figure 6.17, the result of method I (using only the cluster with 4 'PROBE's as the representation of the gene) is more reasonable. We can see the gene which is grouped into one cluster behaves rather differently from the other 4 'PROBE's, and it looks like one outlier. In all, we prefer doing hierarchical clustering before building the mixed effect models.

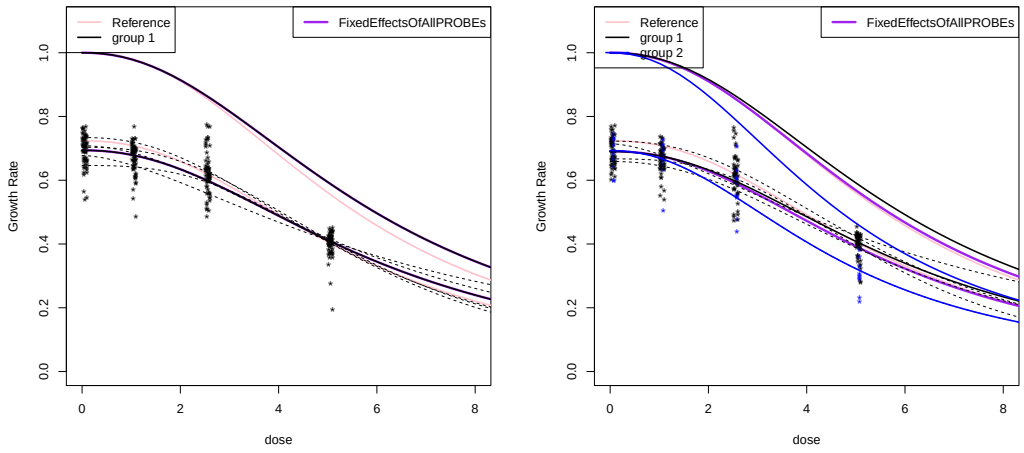

Figure 6.17: The left one is for the gene which is categorized as 'Growth Inhibiting and Weakening the compound effect' ('IW') by both methods. The right one is for one gene which is categorized as 'IW' by method I but is categorized as 'NN' (Growth Neutral and No Interaction) by method II. Black and red curves are the growth rate curves for each cluster. The purple curve is the growth rate curve by method II. The pink curve is the reference. Growth rate curves with Kmax = 1 are adjusted growth rate curves.

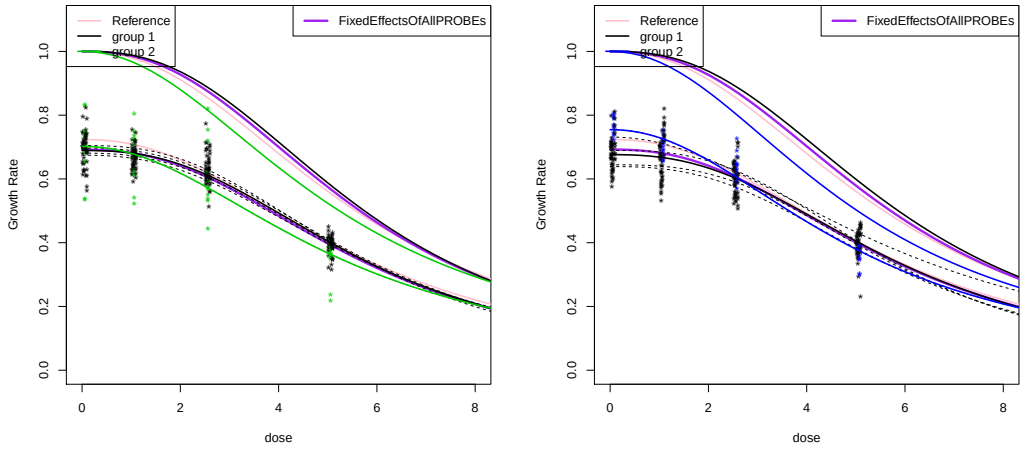

Figure 6.18: The right one is for one gene which is categorized as 'IW' by method I but is categorized as 'NN' (Growth Neutral and No Interaction) by method II. Black and red curves are the growth rate curves for each cluster. The purple curve is the growth rate curve by method II. The pink curve is the reference. Growth rate curves with Kmax = 1 are adjusted growth rate curves.

6.11 Discussion

In this chapter, we discussed the data analysis of one novel experiment, applying hierarchical clustering, nonlinear mixed effect models and hypothesis testing. As this kind experiment is quite new, such analysis is not quite mature. The scientists are not sure about how to deal with the multiple 'PROBE's for each gene. Each 'PROBE' works on a different location of the gene. Their efficiency may not be the same and they may give a different answer, as we have seen in those plots in section 6.10. Hierarchical clustering is proposed to deal with the multiple 'PROBE's for one gene. The idea is that if by hierarchical clustering, the s 'PROBE's for one gene are grouped into one cluster or into two clusters where one cluster contains $s-1$ 'PROBE's, then it is safe to use the big cluster as the representation of the gene. Hence, it is reasonable to build mixed effect model for this cluster with the fixed effect as that of the corresponding gene and the random effect as that of each individual 'PROBE'. However, the drawback of hierarchical clustering is: we do not always get a big cluster. Instead, we could get several small clusters. For example, for one gene with 5 'PROBE's, by hierarchical clustering, we may get 2 clusters, one containing 3 'PROBE's and the other containing 2 'PROBE's. Even worse, we may get 3 clusters, two containing 2 'PROBE's each and one containing 1 'PROBE'. If each cluster gives a different answer about the effects on the cell growth rate with and without the compound, it is difficult to make conclusion about the corresponding gene. More information is needed from the scientists to deal with such cases.

Nonlinear models are used in the analysis. While they fit the data quite reasonably, the convergence issues of nonlinear model exist. Even when it does converge, different start values could give relatively different coefficient estimates and different covariance matrix estimates, which could make some corresponding hypothesis testing unstable. In addition, there are issues about likelihood ratio tests on nonlinear mixed effects models. Right now we treat the likelihood ratio tests on nonlinear mixed effects models the same as on fixed effect models. Improvement about this part could improve the whole data analysis.

Chapter 7

Summaries and Future Research

In this thesis, we brought a new perspective on analyzing microarray data: categorizing the signals in microarray data into three classes - specific signal, nonspecific signal and spurious signal. We discussed one enriched method (enriched PCA-LDA) to highlight the specific signal and weaken the spurious signal. We also showed this method can separate different signals and improve the performance of classification and prediction comparing to some other methods. But as in the weighting procedure, we works with each gene separately, the correlation between genes is ignored. In future, we'd like to consider it. In addition, we would like to try our method on more data, too.

In chapter 5, a theoretical proof about the convergence of the stochastic approximation in conditional t was given. We proved that each element of the vector converges in probability under some reasonable assumptions. However, it would be better if we could prove the vector converges in probability, which will be in our future work.

In chapter 6, we applied hierarchical clustering and nonlinear mixed effect models to analyze 'PROBE' data, in order to determine the effect of each gene on the cell growth rate and the interaction effect between each gene and some compound. We got good results. However, by hierarchical clustering, we could get several small groups of the 'PROBE's. For example, suppose one gene has 6 'PROBE's and by hierarchical clustering, 3 genes are grouped into one class and the other 3 genes are grouped into another class. Then by the nonlinear mixed effect model, each class gives a different answer about the effects on the cell growth rate and the interaction effects. Then it is difficult to get conclusion about the corresponding gene, as theoretically, all 6 'PROBE's can shut down the same gene, they should give the same answer, which will then be used as the answer for the corresponding gene. Right now, we ignore those genes when interpreting the results. We do

hope the scientists could give us some idea about how to deal with those cases. In addition, while nonlinear model goes well with the data, its convergence is one issue. Even when it does converge, different start values could give different coefficient estimates and covariance matrix estimates, which could give opposite answers in hypothesis testing occasionally. We would like to work on this part in our future work, too.

References

[1] Amaratunga, D., Cabrera, J. (2003) Exploration and Analysis of DNA Microarray and Protein Array Data, New York: John Wiley.

[2] Amaratunga, D., Cabrera, J. (2009). A conditional t suite of tests for identifying differentially expressed genes in a DNA microarray experiment with little replication. *Statistics in Biopharmaceutical Research*, **1**, 26–38.

[3] Amaratunga, D., Cabrera, J., Cherckas, Y., Lee, Y. (2012). Ensemble classifiers. *IMS Collections: Contemporary Developments in Bayesian Analysis and Statistical Decision Theory_A Festchrift for William E. Strawderman*, **8**, 235–246.

[4] Amaratunga, D., Cabrera, J., Kovtun, V. (2008). Microarray learning with ABC. *Biostatistics* **9**(1), 128–136.

[5] Amaratunga, D., Cabrera, J., Lee, Y. (2008). Enriched random forests. *Bioinformatics*, **24**(18), 2010–2014.

[6] Bair, E., Hastie, T., Paul, D., Tibshirani, R. (2006). Prediction by supervised principal components. *Journal of the American Statistical Association*, **101**(473), 119–137.

[7] Becker, N., Toedt, G., Lichter, P., Benner, A. (2011). Elastic SCAD as a novel penalization method for SVM classification tasks in high-dimensional data. *BMC bioinformatics*, **12**:138.

[8] Belhumeur, P.N.; Hespanha, J.P.; Kriegman, D.J. (1997) Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection; *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(7), 711-720.

[9] Benjamini, Y., Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society, Series B* **57**(1), 289–300.

[10] Benjamini, Y.; Hochberg, Y. (2000) On the adaptive control of the false discovery rate in multiple testing with independent statistics; *Journal of Educational and Behavioral Statistics*, **25**, 1, 60-83.

[11] Benjamini, Y.; Krieger, A.M.; Yekutieli, D. (2006) Adaptive linear step-up procedures that control the false discovery rate; *Biometrika*, **93** (3), 491-507.

[12] Benjamini, Y.; Liu, W. (1999) A step-down multiple hypotheses testing procedure that controls the false discovery rate under independence; *Journal of Statistical Planning and Inference*, **82**, 163-170.

[13] Benjamini, Y.; Yekutieli, D. (2001) The control of the false discovery rate in multiple testing under dependency; *The Annals of Statistics*, **29**, 1165-1188.

[14] Blum, J.R., (1954) Multidimensional Stochastic Approximation Methods; *The Annals of Mathematical Statistics*, **25**(4), 737-744.

[15] Blum, J.R., (1954) Approximation Methods Which Converge with Probability One; *The Annals of Mathematical Statistics*, **25**, 382-386.

[16] Bo, T.; Jonassen, I. (2002) New feature subset selection procedures for classification of expression profiles; *Genome Biology*, **3**(4), RESEARCH0017.

[17] Bourgon, R., Gentleman, R., Huber, W. (2010). Independent filtering increases detection power for high-throughput experiments. *Proc Natl Acad Sci U S A*, **107**(21), 9546–9551.

[18] Bradley, P. S., Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines. In: J. Shavlik (editor), *Machine Learning Proceedings of the Fifteenth International Conference(ICML 1998)*, San Francisco, California: Morgan Kaufmann, pp. 82–90.

[19] Breheny, P., Huang, J. (2011). Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The Annals of Applied Statistics* **5**(1), 232–253.

[20] Cabrera, J., Devas, V., Fernholz, L.T. (2005) Target Estimation for the Logistic Regression Model; *Joural of Statistical Computation and Simulation*, **75**(2), 121-140.

[21] Cabrera, J., Fernholz, L.T., (1999) Target Estimation for Bias and Mean Square Error Reduction; *The Annals of Statistics*, **27**(3), 1080-1104.

[22] Cabrera, J., Hu, I., (2001) Algorithms for Target Estimation using Stochastic Approximation; *Statistics on the Internet.*

[23] Cabrera, J., Fernholz, L.T. (2004) Multivariate Targeting with Applications to Ellipse Estimation; *Journal of Statistical Planning and Inference*, **122**, 79-94.

[24] Cabrera, J., Watson, G.S., (1997) Simulation Methods for Mean and Median Bias Reduction in Parameteric Estimation; *Journal of Statistical Planning and Inference*, **57**(1), 143-152.

[25] Chen, L.; Liao, H.M.; Ko, M.; Lin, J.; Yu; G. (2000) A new LDA-based face recognition system which can solve the small sample size problem; *Pattern Recognition*, **33**, 1713-1726.

[26] Dudoit, S.; Fridlyand, J.; Speed, T.P. (2002) Comparison of discrimination methods for the classification of tumors using gene expression data; *Journal of the American Statistical Association* **97** (457), 77-87.

[27] Dvoretzky, A. (1956) On Stochastic Approximation; *Proc. Third Berkeley Symp. Math. Statist. Prob.*, **1**, 39-55. University of California Press.

[28] Fan, J., Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* **96**(456), 1348–1360.

[29] Fernholz, L.T. (1997) Target Estimation and Implications to Robustness. In: L_1-Statistical Procedures and Related Topics; *IMS Lecture Notes, Monograph Series*, **31**, 363-372.

[30] Friedman, J.H. (1989) Regularized discriminant analysis; *Journal of the American Statistical Association*, **84**(405), 165-175.

[31] Friedman, J., Hastie, T., Tibshirani, R., (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, **33**(1), 1–22.

[32] Fung, G. M., Mangasarian, O. L. (2001). Proximal support vector machine classifiers. In: F. Provost & R. Srikant (editors), *Proceedings KDD-2001: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* . San Francisco, CA, New York: Asscociation for Computing Machinery, pp. 77–86. ftp://ftp.cs.wisc.edu/pub/ dmi/techreports/01-02.ps.

[33] Fung, G. M., Mangasarian, O. L. (2004) A feature selection Newton method for support vector machine classification. *Computational Optimization and Applications* **28**(2), 185–202.

[34] Furey, T.S.; Cristianini, N.; Duffy, N.; Bednarski, D.W.; Shummer, M.; Haussler, D. (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data; *Bioinformatics*, **16** (10), 906-914.

[35] Ge, Y., Dudoit, S., Speed, T.P. (2003) Resampling-based multiple testing for microarray data analysis; *TEST*, **12**(1), 1-77.

[36] Genovese, C.R.; Roeder, K.; Wasserman, L. (2006) False discovery control with p-value weighting; *Biometrika*, **93** (3), 509-524.

[37] Golub, T.R.; Slonim, D.K.; Tamayo, P.; Huard, C.; Gaasenbeek, M.; Mesirov, J.P.; Coller, H.; Loh, M.L.; Downing, J.R.; Caligiuri, M.A.; Bloomfield, C.D.; Lander, E.S. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring; *Science*, **286**, 531-537.

[38] Guyon, I.; Weston, J.; Barnhill, S.; Vapnik, V. (2002) Gene selection for cancer classification using support vector machines; *Machine Learning*, **46**, 389-422.

[39] Hall, P., Martin, M.A., (1988). On Bootstrap Resampling and Iteration; *Biometrika* **75**(4), 661-671.

[40] Hastie, T.; Buja, A.; Tibshirani, R. (1995) Penalized discriminant analysis; *The Annals of Statistics*, **23**(1), 73-102.

[41] Hastie, T.; Tibshirani, R.; Buja, A. (1994) Flexible discriminant analysis by optimal scoring; *Journal of the American Statisitcal Association*, **89**(428), 1255-1270.

[42] Hastie, T., Tibshirani, R., Friedman, J. (2001). Elements of Statistical Learning: Data Mining, Inference and Prediction, Springer-Verlag, New York, 2001.

[43] Hochberg, Y. (1988) A sharper Bonferroni procedure for multiple tests of significance; *Biometrika*, **75**, 800-803.

[44] Holm, S. (1979) A simple sequentially rejective multiple test procedure; *Scan. J. Stat.*, **6**, 65-70.

[45] Hong, Z.; Yang, J. (1991) Optimal discriminant plane for a small number of samples and design method of classifier on the plane; *Pattern Recognition*, **24**(4), 317-324.

[46] Huang, R.; Liu, Q.; Lu, H.; Ma, S. (2002) Solving the small sample size problem of LDA; *Proceedings of 16th International conference on pattern recognition*, **3**, 29-32.

[47] Khan, J.; Wei, J.S.; Ringnér, M.; Saal, L.H.; Ladanyi, M.; Westermann, F.; Berthold, F.; Schwab, M.; Antonescu, C.R.; Peterson, C.; Meltzer, P.S. (2001) Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks; *Nature Medicine*, **7**(6), 673-679.

[48] Li, H.; Jiang, T.; Zhang, K. (2003) Efficient and robust feature extraction by maximum margin criterion; *Advances in Neural Information Processing Systems*, **16**, 97-104.

[49] Li, H.; Zhang, K.; Jiang, T. (2005) Robust and Accurate Cancer Classification with Gene Expression Profiling; *Proceeding CSB '05 Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference*, 310-321.

[50] Lu, Y.; Tian, Q.; Sanchez, M.; Wang, Y. (2005) Hybrid PCA and LDA analysis of microarray gene expression data; *Computational Intelligence in Bioinformatics and Computational Biology, 2005. CIBCB '05. Proceedings of the 2005 IEEE Symposium*, DOI: 10.1109/CIBCB.2005.1594942

[51] Martin, M.A. (1992) On the Double Bootstrap; *Computing Science and Statistics*, 73-78.

[52] Nguyen, D.V.; Rocke, D.M. (2002) Tumor classification by partial least squares using microarray gene expression data; *Bioinformatics*, **18**, 39-50.

[53] Raghavan, N., De Bondt, A. M.I.M., Talloen, W., Moechars, D., Göhlmann, H.W.H., Amaratunga, D. (2007). The high-level similarity of some disparate gene expression measures. *Bioinformatics*, **23**(22), 3032–3038.

[54] Reiner, A.; Yekutieli, D.; Benjamini, Y. (2003) Identifying differentially expressed genes using false discovery rate controlling procedures; *Bioinformatics*, **19** (3), 368-375.

[55] Robbins, H., Monro, S., (1951) A Stochastic Approximation Method; *The Annals of Mathematical Statistics*, **22**(3), 400-407.

[56] Romano, J.P.; Sharkh, A.M.; Wolf, M. (2008) Control of the false discovery rate under dependence using the bootstrap and subsampling; *Test*, **17**, 417-442.

[57] Sacks, J. (1958) Asymptotic Distributions of Stochastic Approximation Procedures; *The Annals of Mathematical Statistics*, **29**, 373-405.

[58] Schena, M.; Shalon, D.; Davis, R.W.; Brown, P.O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray; *Science*, **270**, 467-470.

[59] Sharma, A., Paliwal, K. K., (2008). Cancer classification by gradient LDA technique using microarray gene expression data. *Data & Knowledge Engineering*, **66**, 338–347.

[60] Slonim, D.; Tamayo, P.; Mesirov, J.; Golub, T.; Lander, E. (2000) Class prediction and discovery using gene expression data; *Proceedings of the 4th Annual International Conference on Computational Molecular Biology (RECOMB) Universal Academy Press, Tokyo, Japan*, 263-272.

[61] Smyth, G.K. (2004) Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments; *Statistical Applications in Genetics and Molecular Biology* Vol. 3: Iss. 1, Article 3. DOI: 10.2202/1544-6115.1027.

[62] Storey, J.D. (2002) A direct approach to false discovery rate; *J. R. Statist. Soc. B*, **64** (3), 479-498.

[63] Storey, J.D. (2003) The positive false discovery rate: a Bayesian interpretation and the q-value; *The Annals of Statistics*, **31** (6), 2013-2035.

[64] Storey, J.D.; Taylor, J.E.; Siegmund, D. (2004) Strong control, conservative point estimation and simultaneous conservative consistency of false discovery rates: a unified approach; *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **66** (1), 187-205.

[65] Storey, J.D.; Tibshirani, R. (2001) Estimating false discovery rates under dependence, with applications to DNA microarrays; *Technical Reports*.

[66] Storey, J.D.; Tibshirani, R. (2003). Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences of the United States of America*, **100**(16), 9440–9445.

[67] Swets, D.L.; Weng, J. (1996) Using Discriminant Eigenfeatures for Image Retrieval; *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**(8), 831-836.

[68] Talloen, W., Clevert, D., Hochreiter, S., Amaratunga, D., Bijnens, L., Kass, S., Göhlmann, H. W. H., (2007). I/NI-calls for the exclusion of non-informative genes: a highly effective filtering tool for microarray data. *Bioinformatics* **23**(21), 2897–2902.

[69] Tian, Q.; Barbero, M.; Gu, Z.; Lee, S.H. (1986) Image classification by the Foley-Sammon transform; *Optical Engineering*, **25**(7), 834-840.

[70] Tibshirani, R. (1996) Regression shrinkage and selection via the Lasso, *Journal of the Royal Statistical Society, Series B (Methodological)*, **58**(1), 267-288.

[71] Tusher, V.G., Tibshirani, R., Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences* **98**(9), 5116–5121.

[72] Wang, L., Zhu, J., Zou, H. (2006). The doubly regularized support vector machine. *Statistica Sinica* **16**, 589–615.

[73] Westfall, P. H.; Young, S. S. (1993) Resampling-based multiple testing: Examples and methods for p-value adjustment; *John Wiley & Sons*.

[74] Wolfowitz, J. (1952) On the Stochastic Approximation Method of Robbins and Monro; *The Annals of Mathematical Statistics*, **23**(3), 457-461.

[75] Yekutieli, D. (2008) False discovery rate control for non-positively regression dependent test statistics; *Journal of Statistical Planning and Inference*, **138** (2), 405-415.

[76] Yekutieli, D.; Benjamini, Y. (1999) Resampling-based false discoverty rate controlling multiple test procedures for correlated test statistics; *Journal of Statistical Planning and Inference*, **82**, 171-196.

[77] Yekutieli, D. (2012) Hierarchical false discovery rate-controlling methodology; *Journal of the American Statistical Association*, **103**(481), 309-316.

[78] Yu, H.; Yang, J. (2001) A Direct LDA Algorithm for High-dimensional Data - with Application to Face Recognition; *Pattern Recognition*, **34**, 2067-2070.

[79] Zhang, C.H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, **38**(2): 894–942.

[80] Zou, H., Hastie, T., (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, **67**(2), 301–320.

List of Figures

List of Tables