EMPIRICAL BAYES ANALYSIS FOR DETECTING DIFFERENTIAL EXPRESSION IN MICROARRAYS

EMPIRICAL BAYES ANALYSIS FOR DETECTING DIFFERENTIAL EXPRESSION IN MICROARRAYS

By

YING WANG, B.Sc., B.Eng.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Science

McMaster University

MASTER OF SCIENCE (2008) McMaster University

(Mathematics and Statistics) Hamilton, Ontario

TITLE: Empirical Bayes Analysis
for Detecting Differential Expression in Microarrays

AUTHOR: Ying Wang, B.Sc (University of Toronto)

SUPERVISOR: Professor Angelo J. Canty

NUMBER OF PAGES: xiii, 71

Acknowledgements

First I would like to sincerely thank my supervisor, Dr. Angelo Canty, for making numerous helpful suggestions and feedback during the course of this work. This thesis would not be possible without his guidance and support.

I am grateful to the members of my Master thesis committee, Dr. Roman Viveros-Aguilera and Dr. Lehana Thabane, for their valuable suggestions and comments. I would also like to thank all faculty members in my department for their academic guidance. Special thanks also go to Dr. Shaheena Bashir and fellow graduate students in my department for their friendship and advice.

I am also grateful to all the people from Dr. Jayne Danska's group at the Hospital for Sick Children in Toronto for providing data. My special thanks will be given to Dr. Evgueni Ivakine for providing the Affymetrix data sets.

I am very much indebted to my husband Chuhong Fei and my son Ryan for their love and encouragement during years of my schooling. I also want to thank my parents for their support throughout my study.

Financial support for this thesis, in the form of Research Assistantships from Genome Canada and NESERC is gratefully appreciated.

Contents

List of Tables

List of Figures

Abstract

The purpose of gene expression microarray studies is to identify differentially expressed genes. Due to the very large number of genes compared to the very small sample size, and the possibility of high level of non-normal random noise, traditional hypotheses tests cannot be used directly. In this thesis, we applied parametric and nonparametric empirical Bayes methodologies to test the hypotheses of differential expression in a real microarray data set from a study of Type 1 Diabetes and some other simulated data sets. In our real data, we saw some problems of applying parametric empirical Bayes (in terms of **R** software called **EBarrays**; nonparametric empirical Bayes method implemented in the **R** packaged **Siggenes** also has problems in detecting differential expression in real data when some extreme patterns show up in the permutation matrix. We implemented a new function called **EBayes** based on Efron's idea of nonparametric empirical Bayes method. **EBayes** performs much better than other empirical Bayes methodologies in dealing with real data. Furthermore, the results of simulated data show that the new function **EBayes** are comparable to **Siggenes EBAM** function.

Chapter 1

Introduction

Microarray experiments are very important in investigating biological phenomena. A wide variety of techniques and algorithms exist for analyzing and extracting information from microarrays. This chapter focuses on nucleotide-based microarrays, one of the most popular types, and provides a simple overview of the steps involved in analyzing these experiments, the important algorithms used today, and the areas of active research.

1.1 Genetic Background

It is well known that there are about approximately ten to the 14th power (10^{14}) cells in the human body. The nucleus of almost every cell comprises the complete human genome. The human genome is the blueprint for all cellular structures and activities in the human body. It consists of 23 pairs of chromosomes. In each pair, one chromosome comes from the mother and the other from the father. Chromosomes are the organized

form of Deoxyribonucleic Acid (DNA) found in cells. They contain a single continuous piece of double-stranded DNA, which contains many genes, regulatory elements and other intervening nucleotide sequences. Each strand of a DNA molecule is built up by a sequence of the bases: Adenine (A), Cytosine (C), Guanine (G) and Thymine (T). Watson and Crick (1953) first proposed the double-helix spatial structure of the DNA. According to the Watson-Crick base pairing rule, the bases are paired so that an A in one strand can only bind to a T in the other, and a C can only bind to a G. The two strands are called complementary, since each strand hence holds the same sequence information. Some segments of the DNA sequence contain genetic information and are loosely called genes. Figure 1.1 shows the structure of the DNA double helix.

The chain constructed from one gene forms a large cellular molecule called a protein. Proteins are the structural components of cells and tissues and perform many key functions of biological systems. Tumor cells differ from normal cells and medically treated cells differ from those untreated cells. The production of proteins is controlled by genes. The extent to which a gene is used to produce proteins is known as *gene expression*. It is a multiple-step process that begins with the "transcription". During transcription, a single strand of messenger ribonucleic acid, or mRNA, is copied from the DNA segment coding the gene. After transcription, mRNA is used as a template to assemble a chain of amino acids to form the protein, this is known as "translation". Transcription and translation are two principal stages involved in protein productions, it is illustrated in the schematic of Figure 1.2.

There are several techniques available for measuring gene expression, such as serial analysis of gene expression (SAGE), cDNA library sequencing, differential

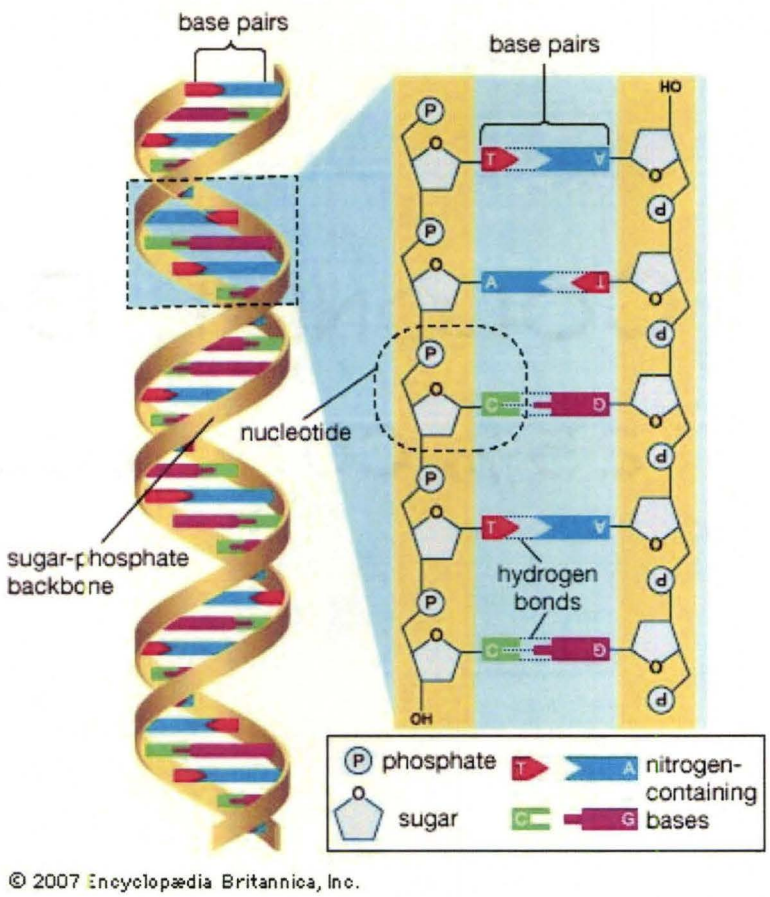

Figure 1.1: *DNA: Molecule. Online Art. Encyclopedia Britannica Online. Graphics from http://www.britannica.com. Encyclopedia Britannica online DNA molecule images. Image Courtesy: Encyclopedia Britannica, Inc.*

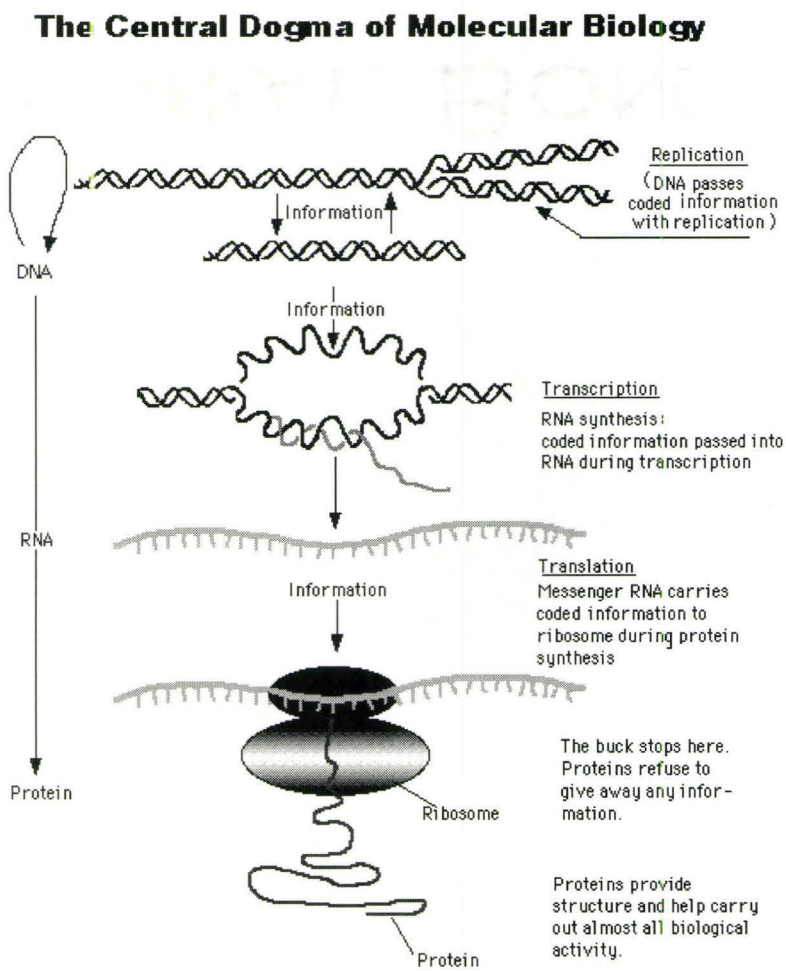

Figure 1.2: *The Central Dogma of Molecular Biology.*

Graphics from http://www.cbs.dtu.dk. Image Courtesy: Dr. Dave Ussery.

display, cDNA subtraction, multiplex quantitative RT-PCR, and gene expression microarrays. Following the Central Dogma of Molecular Biology, if the assumption behind DNA microarrays holds–that most of the mRNA is translated into proteins, then the function of a cell also can be investigated by measuring the mRNA levels. Here we will focus our interest on the analysis of DNA microarrays.

There are several microarray technologies. The main types of gene expression assays are: spotted cDNA arrays, short oligonucleotide arrays (Affymetrix), long oligonucleotide arrays (Agilent Inkjet), and fibre optic arrays (Illumina). One prevalent approach involves the use of high-density oligonucleotide arrays, the most widely used oligonucleotide array type is the Affymetrix GeneChip (for brevity Affy). In Affy arrays, expression of each gene is measured by comparing hybridization of the sample mRNA to a set of probes, composed of $11 - 20$ pairs of oligonucleotides, each of length 25 bases. The first type of probe in each pair is known as perfect match (PM) and is taken from the gene sequence. The second type is known as mismatch (MM) and is created by changing the middle $(13th)$ base of the PM sequence to reduce the rate of specific binding of mRNA for that gene. The goal of MMs is controlling for experimental variation and nonspecific binding of mRNA from other parts of the genome.

Oligonucleotide arrays are well discussed by Lockhart et al. (1996); and Affymetrix (1999) gives details on Affy arrays. By an llustration of Affymetrix GeneChip in Figure 1.3, we can see that an RNA sample is prepared, labelled with a fluorescent dye, and hybridized to an array. Arrays are then scanned, and images are produced and analyzed to obtain a fluorescence intensity value for each probe, measuring hybridization for the corresponding oligonucleotide. For each gene, or probe set, the typical

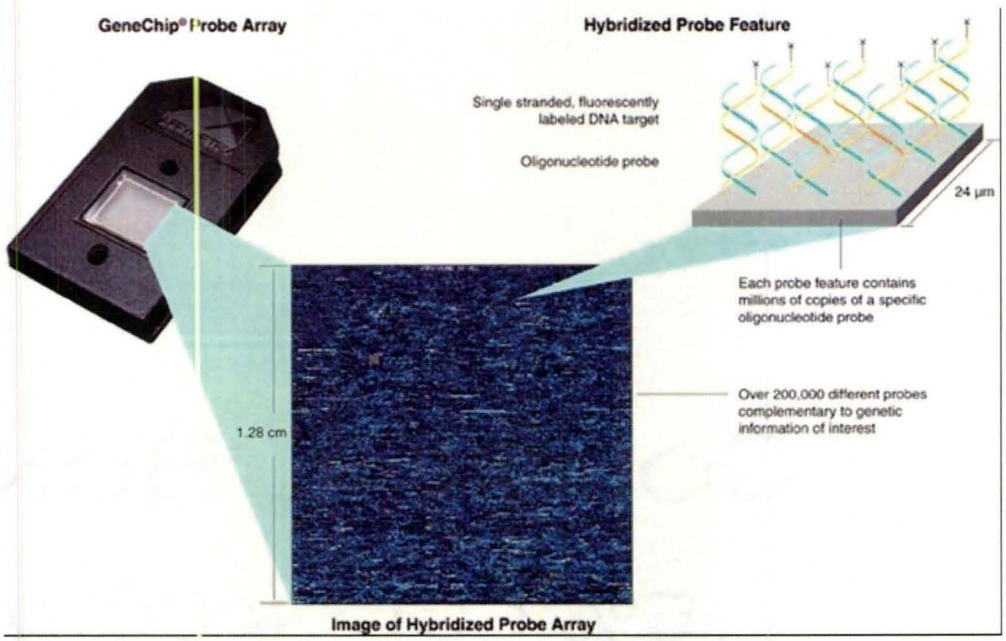

Figure 1.3: *An Illustration of Affymetrix GeneChip.*

Graphics from http://www.jyi.org. Image Courtesy: Affymetrix.

output consists of two vectors of intensity readings, one for PMs and one for MMs. Then different approaches of data analysis for deriving probe-set summaries that best reflect the level of expression of the corresponding genes can be made.

1.2 Overview of Microarray Data Analysis

Over the past decade there have been two dramatic changes in microarray data analysis. First, with the advance of DNA hybridization microarray technologies nowadays, it is possible to simultaneously assess the expression levels of tens of thousands of genes. So studies of single genes are being replaced by studies that probe many genes simultaneously. It means our analysis also changes to dealing with a family

of tens of thousands of t-tests simultaneously instead of a two-sample t-test or some other "t-like" tests. Second, different types of biological information, such as genomic alterations, mRNA levels, and protein levels, are being combined together in an attempt to give a comprehensive view of biological processes. So for the application of high-throughput measurement of gene expression, various groups have demonstrated that the use of modern statistical methodology can substantially improve accuracy and precision of the gene expression measurements, relative to ad hoc procedures introduced by designers and manufacturers of the technology.

To adapt to these changes, computational statisticians are engaged in developing new methodologies for analysis of genome-wide data sets. Their analysis will serve as a guide for analyzing microarray data and for computing unbiased estimates of relative gene expression.

1.2.1 General Steps in Microarray Data Analysis

We can treat microarray analysis as a series of sequential steps because the output of one step feeds into the next step, and each step converts one type of data into another by various algorithms and software packages. Generally, there are six steps of data processing: quantization, background correction, mismatch adjustment, normalization, summarization and statistical analysis (Boutros, 2007). The first step - quantization - is an image analysis. The next four steps - background correction, mismatch adjustment, normalization, and summarization - are preprocessing steps to produce gene expression measurements/values. The last step - statistical analysis - is to identify the DNA sequence variants in specific genes or regions of the human

genome that are associated with phenotypes of interest such as disease.

Step 1: Quantization

The first step in microarray data analysis is called "quantization" step because it convert an image into a series of numbers. A microarray experiment is performed as follows: first the labelled mRNA is hybridized to the array; then the array is washed to remove the excess or weakly bound mRNA; next the microarray is excited by a laser and finally the array is scanned. So we will get a picture of the array in which high binding will be visible as white spots on a black background. Therefore, quantization is a mandatory step in all microarray analyses. It takes the raw image as an input, and converts it to estimates of signal in a spot (foreground signal, FG) and of nonspecific signal in the surrounding regions to a spot (background signal, BG). In this thesis, we use the default Affymetrix quantization algorithm.

Step 2: Background Correction

As we have discussed, a preprocessing procedure includes four steps: background correction, mismatch adjustment, normalization and summarization. They are used to produce an expression measurement. At each stage of this procedure, numerous methods have been proposed for GeneChip arrays.

Background correction is also referred to as signal adjustment. The scanning of arrays results in optical and background noise affecting pixel intensities. Because each spot in a microarray has both specifically bound DNA and non-specific signal, a slight signal could be seen in the area that is in between spots on some array images. So background information is difficult to obtain. Background corrections are used to remove a non-specific signal that arises from non-specific hybridization, the slide itself, or coatings or other materials on the slide. Many studies have demonstrated

that careful removal of this signal can significantly increase the signal-to-noise ratio of a microarray experiment.

Numerous background correction methods have been proposed. For Affy data, background correction is done by a maximum likelihood deconvolution of the probe intensity into an exponential signal and normal noise. This procedure is a part of the "RMA" (Robust Multi-chip Average) procedure in **R** package **Affy**(Irizarry, Gautier, & Cope, 2003). So we will apply RMA deconvolution background correction in our project.

Step 3: Mismatch Adjustment

As we discussed in Section 1.1, Affymetrix GeneChip expression arrays usually contain two types of sequences: "perfect match" (PM) and "mismatch"(MM). PM matches an mRNA transcript exactly. MM is almost the same as PM except that MM has a 1 mutation in the center of its sequence. This mutation will prevent binding of the actual transcript, but will allow any "non-specific hybridization" still to occur. So MM can be used to remove non-specific hybridization. If the mismatch sequence performs as expected, this process can dramatically reduce the noise in an array experiment.

The biggest problem of mismatch adjustment is that this process rarely performs as expected. One Affymetrix analysis software MicroArray Suite (MAS) in version 4.0 calculated the signal as PM-MM, but the real analysis shows that there are about 30% of data points that have negative values of signal. Later Irizarry, Hobbs, et al. (2003) found that MM data only added noise but contributed very little to signals. Therefore, RMA totally ignores all MM probes and only utilizes the PM probes.

Step 4: Normalization

Normalization is an important step in microarray data analysis. It is designed to remove variation of non-biological noise and systematic artifacts within or between arrays so that their values can be comparable. Sources of obscuring variation include variation introduced during the process of sample preparation, during the manufacture of the array, during the hybridization of the sample on the array, and during the scanning and analysis of fluorescent intensity after hybridization. The obscuring sources of variation can have many different effects on data, unless arrays are appropriately normalized.

Various normalization methods have been proposed, such as constant normalization, contrasts normalization, invariant set normalization, loess normalization, qspline and quantile normalization. Yang (2006) compared a number of commonly used and state-of-art normalization methods in microarray analysis, such as Robust Multi-chip Average (RMA), MAS5.0, GCRMA, PLIER and dChip. RMA has quantile normalization as build in method. Based on the assumption that the distribution of expression levels is constant across chips, quantile normalization assumes that the chips have a common distribution of intensities, so they may be transformed to produce similar distributions. Therefore, after quantile normalization, all probe sets will have same box plots across the chips. RMA uses quantile normalization of the background-corrected PM probes because it provides a fast method to normalize multiple chips.

Step 5: Summarization

A single gene might be represented by many sequences on an array. Different sequences might represent different parts of the gene, or they may be replicates. As

the last step in pre-processing, summarization is where multiple probe intensities within a probe set are combined to produce an expression value.

Commonly discussed summarization methods include Average Difference (Avgdiff) summarization, median polish summarization, MAS summarization, and Li and Wong (2001) summarization. For Affymetrix arrays, the RMA algorithm always uses median polish method to combine the signals from multiple probes together. Median polish is a data analysis technique (more robust than ANOVA) for examining the significance of various factors in a multi-factor model. In our case there are 2 factors: the array and the probe. The estimate of the arroy coefficient is defined to be the expression for the probe set or the array. This procedure is called "robust" because it is relatively insensitive to outliers. We will apply median-polish as part of RMA preprocessing to summarize our genome data in this project.

Step 6: Statistical Analysis

In microarray data analysis, our ultimate goal is to identify genes that change between experimental conditions by appropriate statistical analysis methods. For example, in our study, we want to identify genes that have different expression in strains of mice that are resistant or susceptible to Type 1 Diabetes (T1D) out of the bulk of the microarray data.

A serious statistical problem in microarray experiments is that the sample size is too small compared to the big number of sample dimensions, which may cause difficulties in data reduction and simultaneous inference. Several techniques have been tried to solve these problems, such as principle component analysis (PCA), empirical Bayes analysis, clustering analysis and Significant Analysis of Microarrays (SAM). In this project, parametric and nonparametric empirical Bayes will be discussed and

	Accept Null	Reject Null	Total
Null True	U	V	G_0
Alternative True	T	S	G_1
Total	W	R	G

Table 1.1: *Possible outcomes from m hypothesis tests*

we will focus our study on nonparametric empirical Bayes. A simple nonparametric empirical Bayes model proposed by Efron et al. (2001) will guide the efficient reduction of the data to a single summary statistic per gene, and also to make simultaneous inferences concerning differentially expressed genes. We will discuss three different variations of nonparametric empirical Bayes based on Efron's idea in detail, and then apply them to a real microarray data set and some simulated data sets.

1.2.2 Multiple-Testing Problem and Bayesian Methods

In microarray data analysis, multiplicity makes identification of differentially expressed genes very hard. In multiple hypotheses testing, suppose there are G independent test statistics with level α, then $P(\text{at least one falsely reject}) = 1 - (1-\alpha)^G$, which will rise with G. The number of probe sets (G) in an Affy chip is large. For example, in our Affy data sets, the number of probe sets is $G = 12488$. So the probability of type I error in this multiple hypothesis testing is very close to 1.

Take an example of multiplicity test from Table 1.1, where V is the number of Type I errors (false positives), T is the number of Type II errors (false negatives), and $R = V + S$ is the total number of significant hypotheses. In order to measure the errors incurred in multiple hypothesis testing and control the Type I error, Benjamini

and Hochberg (1995) first proposed the False Discovery Rate (FDR) as the following:

$$FDR = \mathbf{E}\left(\frac{V}{R}|R > 0\right) \cdot \mathbf{Pr}(R > 0) \tag{1.2.1}$$

Instead of calculating the chance of any false positives $E[V]$, FDR controls the expected proportion of false positives among the number of all genes which have been claimed positive, so it is adaptive to the amount of signal in data. FDR is a new approach to the multiple comparisons problem. Some statisticians believe it is a bridge between traditional statistical thinking and modern problems in data mining and bioinformatics.

Positive False Discovery Rate (pFDR) was proposed by Storey (2002). It is defined as the following:

$$pFDR = E\left(\frac{V}{R}|R > 0\right) \tag{1.2.2}$$

The term "positive" has been added to reflect the fact that we are conditioning on the event that positive findings have occurred. Although the ideas of FDR and pFDR were originally proposed under the assumption that all p values are independent. In more general cases, such as with dependence or in nonparametric situations, it is possible to apply very similar ideas to obtain accurate estimates of pFDR and FDR. For a large class of dependence structure, Storey and Tibshirani (2001) show that the effect of dependence could be negligible if G is large. Therefore, this multiple-hypothesis testing methodology is useful not only in fields like genomic analysis but also in the field of data mining.

Early statistics for microarray experiments include gene specific t-tests and/or permutation methods by Dudoit et al. (2002), as well as maximum likelihood methods,

to find differentially expressed genes. Because microarrays hold information about thousands of genes simultaneously, it is natural to use empirical Bayes approaches. If the overall information about thousands of genes is summarized into prior parameters, their performances will be much superior to the usual t-statistic and similar methods. However, finding appropriate distributional assumptions to produce conjugate priors for the parameters is not easy. We will discuss empirical Bayes approaches (including parametric and nonparametric Bayes) in detail later in Chapter 2.

1.3 Type I Diabetes and Mouse Model Experiment Design

Due to multiple genetic risk factors and currently unknown environmental factors, Type 1 Diabetes (T1D) is a complex, autoimmune-mediated disease. Since T1D is usually diagnosed in children and young adults, it was previously known as juvenile diabetes. In type 1 diabetes, the body does not produce insulin. Insulin is a hormone that is needed to convert sugar (glucose), starches and other food into energy needed for daily life. As Llanos and Libman (1994) found, the incidence of T1D varies widely between populations. It happens $0.7/100,000$ people per year in Peru and $45/100,000$ people per year in Finland. Canada has the third highest rate in the world. In the past 50 years, the incidence of T1D has risen rapidly. Furthermore, T1D cost the Canadian health care system \$1.32 billion in 2002 and is projected to rise to \$1.6 billion by 2010 (Newhook et al., 2004). Therefore, biologists are working hard to define complex genetic contributions to T1D. And computational statisticians are

trying to develop new methodology to analyze genome-wide data sets to get good estimates of relative gene expression.

One genetic region called Idd4 has been shown to affect genetic susceptibility to T1D, so biologists are interested in the Idd4 locus on the mouse genome. By mating two inbred strain mice: non-obese diabetic (NOD) mice and non-obese resistant (NOR) mice, and back crossing the descendants $5 - 10$ generations, biologists get NOD.NOR-Idd4 congenic mice. The NOD.NOR-Idd4 mice have the NOD genome everywhere except at the Idd4 locus where they have the NOR genome. Since studies show 85% female NOD mice get diabetes by 6 months of age while NOR mice are diabetes resistant, although 88% of their genomes are identical to NOD mice, we want to find the difference between NOD.NOR-Idd4 mice and NOD mice in their gene expression profiling.

In this project, we use mice data from Affymetrix GeneChip *MGU74aV2*. The objective of the study is to identify the differentially expressed genes among 12488 probe sets by these Affy chips. The data is processed on two different days to obtain eight arrays: four replicates that include two NOD strains and two NOD.NOR-Idd4 strains on day one, and four replicates that include two NOD stains and two NOD.NOR-Idd4 strains on day two. We will adjust for the day effect on these eight chips first, then apply different statistical analysis to find differentially expressed genes.

1.4 Organization of the Thesis

The objective of this project is to apply parametric and nonparametric empirical Bayes analysis to detect differential expression of genes between NOD mice and NOD.NOR-Idd4 mice, based on Affymetrix GeneChip *MGU74aV2* data sets we have. The thesis is organized as follows. Chapter 2 provides necessary preliminaries of empirical Bayes analysis (including parametric and nonparametric empirical Bayes methodologies) in microarrays. In Chapter 3, we first apply two empirical Bayes methods to a real microarray data set. Then we investigate the problems we found in parametric empirical Bayes analysis on microarrays (based on **R** package named **EBarrays**), and the problems in nonparametric empirical Bayes analysis on microarrays (based on **R** package named **Siggenes**). Then we go through the model from Efron's paper (Efron et al., 2001) and propose three new methods derived from it. Simulation comparisons of these four nonparametric empirical Bayes methods are addressed in Chapter 4. Finally, conclusions are drawn in Chapter 5, and future work will also be discussed.

Chapter 2

Empirical Bayes Analysis in Microarrays

Empirical Bayes (EB) methods have been popular for quite a long time. The earliest work can be traced back to the 1940's by von Mises, but the first major work must be attributed to Robbins (1955). Applying Empirical Bayes approaches to make inferences from microarrays is natural because microarrays hold information about thousands of genes simultaneously. But the sample size is relatively small. Therefore, the amount of information per gene can be relatively low. Efron and Morris (1977) analyzed the so-called *Stein Effect* in Empirical Bayes methods. Roughly speaking, the Stein Effect asserts that estimates can be improved by using information from all coordinates when estimating each coordinate. In microarrays, the data from other genes provide some information about the typical variability in the system. Furthermore, since microarrays hold information about thousands of genes simultaneously, if we summarize the overall information into prior parameters, and combine it with

means and standard deviations at the gene level, their performance will be much superior to the usual t-statistic.

The major difference between parametric and nonparametric empirical Bayes analysis is that the parametric approach specifies a parametric family of prior distributions, but the nonparametric approach leaves the prior completely unspecified. In the following sections, we are going to discuss parametric and nonparametric empirical Bayes methodologies, how to apply them to microarrays, and their disadvantages and advantages.

2.1 Parametric Empirical Bayes Analysis and EM Algorithm

Since the first major work of parametric empirical Bayes analysis is done by Efron and Morris in the 1970s, they have been called the founders of modern empirical Bayes data analysis (Casella, 1985). In microarray data analysis, there are two models quite prevalent: the Gamma-Gamma model and Lognormal-Normal model. These are proposed in Newton and Kendziorski (2003).

2.1.1 The Gamma-Gamma and Lognormal-Normal Models of Parametric Empirical Bayes Analysis

Let Y_{ij} be the expression level of gene i in array j ($i = 1, ..., G$; $j = 1, ..., n_1$, $n_1 + 1, ..., n_1 + n_2 = n$) for a two-condition model structure, where the first n_1 arrays and last n_2 arrays are obtained under the two different conditions. We want to char-

acterize the probability distribution of $\mathbf{Y}_i = (y_{i1}, y_{i2}, ..., y_{in})$. The basic assumption of microarray data analysis is that the majority of genes have unchanged expressions across arrays, so these n samples are exchangeable. Thus, y_{ij} can be treated as independent random deviations from a gene-specific mean values μ_i and they have an observed distribution $f_{obs}(\cdot|\mu_i)$.

Suppose the sample set can be partitioned into two subsets n_1 and n_2 with corresponding mean values μ_1 and μ_2. If the distribution of measured expression is not affected by this grouping, we say that there is equivalent expression (EE_i) for gene i; otherwise, there is differential expression (DE_i). Then we can assume that the gene effects arise independently and identically from a system-specific distribution $\pi(\mu)$, and this allows genes to share information.

If the fraction of differentially expressed genes among all genes is p, then the fraction of equivalently expressed genes is $1 - p$. An EE gene i presents data $\mathbf{Y}_i = (y_{i1}, y_{i2}, \ldots, y_{in})$ which will have a distribution

$$f_0(y_i) = \int \left(\prod_{j=1}^{n} f_{obs}(y_{ij}|\mu) \right) \pi(\mu) d\mu. \tag{2.1.1}$$

Alternatively, if gene i is differentially expressed, the data $\mathbf{y}_i = (\mathbf{y}_{i(1)}, \mathbf{y}_{i(2)})$ will have a distribution

$$f_1(\mathbf{y}_i) = \left(\int \left(\prod_{j=1}^{n_1} f_{obs}(y_{ij}|\mu) \right) \pi(\mu) d\mu \right) \left(\int \left(\prod_{j=n_1+1}^{n_1+n_2=n} f_{obs}(y_{ij}|\mu) \right) \pi(\mu) d\mu \right). \tag{2.1.2}$$

So the marginal distribution of the data is

$$p f_1(\mathbf{y}_i) + (1-p) f_0(\mathbf{y}_i). \tag{2.1.3}$$

By Bayes' rule and known estimates of f_0, f_1 and p, the posterior probability of

differential expression is

$$P(\mathrm{DE}_i|\mathbf{y}_i) = \frac{pf_1(\mathbf{y}_i)}{pf_1(\mathbf{y}_i) + (1-p)f_0(\mathbf{y}_i)} \tag{2.1.4}$$

- **Gamma-Gamma (GG) model**

The GG model supposes that the observation component has a Gamma distribution with shape parameter $\alpha > 0$ and mean value μ, so the scale parameter is $\lambda = \alpha/\mu$. Thus, for measurements $y > 0$,

$$f_{obs}(y|\mu_i) = \frac{\lambda^\alpha y^{\alpha-1} \exp\{-\lambda y\}}{\Gamma(\alpha)} \tag{2.1.5}$$

So the marginal distribution $\pi(\mu_i)$ can be taken to be an inverse Gamma distribution with shape parameter α_0 and scale parameter ν for a fixed α. Therefore, the key density $f_0(\cdot)$ has the form

$$f_0(y_1, y_2, ...y_n) = K\frac{(\prod_{j=1}^n y_j)^{\alpha-1}}{(\nu + \Sigma_{j=1}^n y_j)^{I\alpha+\alpha_0}} \tag{2.1.6}$$

where

$$K = \frac{\nu^{\alpha_0}\Gamma(n\alpha + \alpha_0)}{\Gamma^n(\alpha)\Gamma(\alpha_0)}.$$

- **Lognormal-Normal (LNN) model**

In the LNN model, we assume log-transformed measurements for each gene i have a normal distribution with mean μ_i and common variance σ^2. A conjugate prior for the μ_i is normal distribution with mean μ_0 and variance τ_0^2. So the density $f_0(\cdot)$ for an n-dimensional input becomes Gaussian with mean vector μ_0=$(\mu_0, \mu_0, .. \mu_0)^t$ and exchangeable covariance matrix

$$\mathbf{\Sigma}_n = (\sigma^2)\mathbf{I_n} + (\tau_0)^2\mathbf{M_n},$$

where $\mathbf{I_n}$ is an $n \times n$ identity matrix and $\mathbf{M_n}$ is an $n \times n$ matrix of ones.

Because GG and LNN models both hold the assumption of a constant coefficient of variation (CV), this property can be checked before we do model fitting.

2.1.2 EM Algorithm

In both GG and LNN models, we need to apply maximum (marginal) likelihood method, to estimate unknown parameters (α, α_0, ν) and $(\mu_0, \sigma^2, \tau_0^2)$ respectively. We can estimate those parameters by expectation-maximization (EM) algorithm.

An EM algorithm is used in statistics for finding maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved latent variables. Dempster et al. (1977) first generalized the method and developed the theory behind it. They also explained and gave the name of EM algorithm in this paper. The EM algorithm includes two steps: Expectation (E) step and Maximization (M) step. The E step computes an expectation of the likelihood by including the latent variables as if they were observed; The M step computes the maximum likelihood estimates of the parameters by maximizing the expected likelihood found on the E step. Then the parameters found in the M step will be passed to a new E step. This process will be repeated until the estimates found in E step and M step converge to each other.

With data Y_i governed by a mixture model from Equation (2.1.3), we introduce pattern indicator ϕ_{il}: $\phi_{il} = 1$ when expression pattern on gene i is pattern l; $\phi_{il} = 0$ otherwise. So the complete data log likelihood is

$$l_c(\theta) = \sum_i \left\{ \sum_{k=0}^{g} \phi_{ik} [\log f_k(y_i) + \log(p_k)] \right\}. \tag{2.1.7}$$

In E-step, based on a current estimate θ_0, the expectation given the observed

data amounts to replacing ϕ_{il} with $\hat{\phi}_{il}$. In the M-step, we use the arithmetic mean of $\hat{\phi}_{.,k}$ to estimate p_k, then we can get updated estimates of θ. This process is iterated until successive estimates stabilize.

2.2 Nonparametric Empirical Bayes Analysis in Microarrays

From the previous sections in this chapter, we know that parametric tests may not be valid for microarrays in practice since they have too strong parametric assumptions or large sample justifications. As alternatives, nonparametric statistical methods, such as empirical Bayes method of Efron et al. (2001) and the significance analysis of microarray (SAM) method of Tusher et al. (2001) have been proposed. Those two methods both rely on constructing a test statistic and a so-called null statistic such that the distribution of null statistic could be used to approximate the null distribution of the test statistic. We will focus our work on nonparametric empirical Bayes method in this project. In this section, we will give a brief description of how nonparametric empirical Bayes method works and the general steps to apply this method to microarrays.

2.2.1 Review of Nonparametric Empirical Bayes Analysis in Microarrays

Suppose that Y_{ij} is the expression level of gene i in array j ($i = 1, ..., G$; $j = 1, ..., n_1$, $n_1 + 1, ..., n_1 + n_2 = n$). Suppose that the first n_1 and last n_2 arrays are obtained

under the two different conditions respectively. A general statistical model is

$$Y_{ij} = a_i + b_i x_j + \epsilon_{ij} \tag{2.2.8}$$

where $x_j = 0$ for $1 \leq j \leq n_1$, and $x_j = 1$ for $n_1 + 1 \leq j \leq n_1 + n_2$, and ϵ are independent (but not necessarily identically distributed) random errors with mean 0. Hence a_i and $a_i + b_i$ are the two mean expression levels of gene i under the two conditions respectively. So the hypothesis test to find differentially expressed genes is $H_0 : b_i = 0$ Vs. $H_1 : b_i \neq 0$.

Let the sample means of Y_{ij}'s for gene i under the two conditions be $\bar{Y}_{i(1)} = \frac{\sum_{j=1}^{n_1} Y_{ij}}{n_1}$, $\bar{Y}_{i(2)} = \frac{\sum_{j=n_1+1}^{n_1+n_2} Y_{ij}}{n_2}$, and let s_i be the pooled standard deviation for gene i:

$$s_i = \sqrt{\left(\frac{1}{n_1} + \frac{1}{n_2}\right) \frac{\sum_{j=1}^{n_1}(Y_{ij} - \bar{Y}_{i(1)})^2 + \sum_{j=n_1+1}^{n_1+n_2}(Y_{ij} - \bar{Y}_{i(2)})^2}{n_1 + n_2 - 2}} \tag{2.2.9}$$

Then a reasonable test statistic for assessing differential gene expression is the standard (unpaired) t-statistic: $t_i = \frac{\bar{Y}_{i(2)} - \bar{Y}_{i(1)}}{s_i}$. To reduce the overall variance of the s_i, giving the tests more power on average, Tusher et al. (2001) take a nonparametric approach to this and shrink the s_i toward an adaptively chosen s_0. The modified t-statistic is then

$$Z_i = \frac{\bar{Y}_{i(2)} - \bar{Y}_{i(1)}}{s_i + s_0} \tag{2.2.10}$$

where s_0 is chosen as the percentile of the s_i values that makes the coefficient of variation of Z_i approximately constant as a function of s_i. This has the added effect of dampening large values of z_i that arise from genes whose expression is near zero.

The nonparametric empirical Bayes method proposed by Efron et al. (2001) attempts to avoid highly specified models, relying instead on a simple inference model. Let p_1 be the probability that a gene is affected, $p_0 = 1 - p_1$ be the probability

unaffected, and $f_1(Z)$ be the density of expression Z for affected genes, $f_0(Z)$ the density of Z for unaffected genes, then the mixture density of the two populations is

$$f(Z) = p_0 f_0(Z) + p_1 f_1(Z) \tag{2.2.11}$$

In our situation, we can estimate $f(Z)$ directly from the i-th expression scores Z_i obtained from the Equation (2.2.10). Concentrating on the two-sample case, the null distribution $f_0(Z)$ can be calculated by permuting the group labels, or one can use the bootstrap. Here we will use permutation method because it has a strength in that if the null hypothesis is true, then we can calculate the null distribution.

Applying Bayes' rule to the mixture model in Equation (2.2.11), we can get posterior probabilities $p_1(Z)$ and $p_0(Z)$ as

$$\begin{aligned} p_1(Z) &= 1 - p_0 \frac{f_0(Z)}{f(Z)}, \\ p_0(Z) &= p_0 \frac{f_0(Z)}{f(Z)} \end{aligned} \tag{2.2.12}$$

where $p_1(Z)$ is the posterior probability for differentially expressed genes and and $p_0(Z)$ is the posterior probability for equivalently expressed genes.

Obviously, if we can estimate the value of p_0 and ratio $\frac{f_0(Z)}{f(Z)}$, then the posterior probabilities will be found. One way to estimate this ratio is using their relative densities from the observed score $\{Z_i\}$ and permuted score $\{z_i\}$'s empirical distributions. If we consider values of $\{Z_i\}$ as "success" and values of $\{z_i\}$ as "failures", then with $G = 12488$ genes and $B = 20$ permutations, we can plot $G(1+B) = 12488 \times 21$ total scores on a line, where $G = 12488$ scores from observed $\{Z_i\}$ and $G \times B = 12488 \times 20$ scores from permutated $\{Z_i\}$. So the probability $\pi(Z)$ of a success at point z is given

as

$$\pi(Z) = \frac{f(Z)}{f(Z) + Bf_0(Z)} \tag{2.2.13}$$

so the posterior probability of differentially expressed gene $p_1(Z)$ becomes

$$p_1(Z) = 1 - p_0 \frac{1 - \pi(Z)}{B\pi(Z)} \tag{2.2.14}$$

We can estimate $\pi(Z)$ by logistic regression as follows: first divide the range of the observed and permuted statistics into several equal intervals; then find the frequencies of the observed and permuted values in each interval; finally pass those values to a logistic regression function with a natural spline on degrees of freedom equals 5 to estimate $\pi(Z)$.

Another value we should know to get $p_1(Z)$ is p_0 (the probability that a gene is unaffected). Since the posterior probability of differentially expressed genes $p_1(Z)$ is nonnegative for all Z, it restricts p_0 and p_1 as

$$\begin{aligned} p_1 &\geq 1 - \min_Z \frac{f(Z)}{f_0(Z)}, \\ p_0 &\leq \min_Z \frac{f(Z)}{f_0(Z)} \end{aligned} \tag{2.2.15}$$

Therefore, the upper bound of p_0 is equal to $\frac{f(Z)}{f_0(Z)}$, so we can estimate p_0 from the value of this relative densities.

In summary, the algorithm for nonparametric empirical Bayes analysis for microarrays is the following:

- Step 1

  Compute the Z statistic for observed data values Y by Equation (2.2.10);

- Step 2

  Generate B independent row-wise sign permutations, compute the z statistic for permuted data values in the same way as Z was calculated;

- Step 3

  Apply logistic regression to estimate the probability of success $\pi(Z)$ based on the relative densities of the Z_i and z_i, then get ratio $\frac{f_0(Z)}{f(Z)}$;

- Step 4

  Use the upper bound of $\frac{f(Z)}{f_0(Z)}$ to estimate p_0;

- Step 5

  Find the posterior probability $p_1(Z)$ for each gene from Equation (2.2.14).

Obviously, other variants of this algorithm could be applied to estimate the posterior probabilities but this is the method given by Efron et al. (2001). More details about other variations will be discussed in Chapter 3 and Chapter 4.

2.2.2 False Discovery Rate of Nonparametric Empirical Bayes Analysis

From the development in the Section 2.2.1, we can see that the nonparametric empirical Bayes analysis is very closely related to Benjamini and Hochberg's False Discovery Rate (FDR) criterion. The FDR is the expected proportion of type I error made using a given rejection rule, so Efron defines the *local false discovery rate* as

$$\text{fdr(Z)} = p_0 \frac{f_0(Z)}{f(Z)}. \tag{2.2.16}$$

Thus local fdr(Z) is the *posterior probability* $p_0(Z)$.

Now we need to define estimated FDR ($\widehat{\text{FDR}}$) for a rejection region $c = (-\infty, a) \cup (b, +\infty)$ for our microarray data.

Let $l_o(C)$ be the number of observed test scores in this region, $l_p(C)$ be the number of permuted test scores in this region, and B is the number of permutations, then

$$\widehat{\text{FDR}}_C = \hat{p}_0 \frac{l_p(C)/B}{l_o(C)} \tag{2.2.17}$$

where $\hat{p}_0$ is the estimated probability of unaffected gene. Since we use the upper bound of $\frac{f(Z)}{f_0(Z)}$ to estimate $\hat{p}_0$, it turns out the value of $\widehat{\text{FDR}}_C$ we calculated is slightly conservative.

Chapter 3

Real Microarray Data Analysis by EBarrays, Siggenes and EBayes

In Chapter 2, we discussed the preliminaries of parametric and nonparametric empirical Bayes analysis for detecting differential expression in microarrays. In this chapter, we will investigate the application results of these methodologies in our real microarray data.

3.1 Microarray Data Set

From the introduction of mouse model experiment design in Section 1.3, we know that there are eight microarray chips in our real microarray data sets, and each array has 12,488 individual DNA sequences. The first four chips are taken from male NOD mice and the last four chips are taken from male NOD.NOR-Idd4 mice. Our objective is to detect which genes on locus Idd4 have effects on T1D resistance. After preprocessing

Probe Set Id.	$\mathbf{Y}_1$	$\mathbf{Y}_2$	$\mathbf{Y}_3$	$\mathbf{Y}_4$	$\mathbf{Y}_5$	$\mathbf{Y}_6$	$\mathbf{Y}_7$	$\mathbf{Y}_8$
"93427_at"	$Y_{1,1}$	$Y_{1,2}$	$Y_{1,3}$	$Y_{1,4}$	$Y_{1,5}$	$Y_{1,6}$	$Y_{1,7}$	$Y_{1,8}$
"104748_s_at"	$Y_{2,1}$	$Y_{2,2}$	$Y_{2,3}$	$Y_{2,4}$	$Y_{2,5}$	$Y_{2,6}$	$Y_{2,7}$	$Y_{2,8}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
"92557_f_at"	$Y_{12488,1}$	$Y_{12488,2}$	$Y_{12488,3}$	$Y_{12488,4}$	$Y_{12488,5}$	$Y_{12488,6}$	$Y_{12488,7}$	$Y_{12488,8}$

Table 3.1: *Expression values of* $\boldsymbol{Y}$.

steps of data analysis by RMA function from **Affy** package in **R** software, we have a $12,488 \times 8$ matrix $\mathbf{M}$ of expression values, one value for each gene on each array. Because eight microarray chips are taken from two different days (chips $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{M}_5$, $\mathbf{M}_6$ from day one and chips $\mathbf{M}_3$, $\mathbf{M}_4$, $\mathbf{M}_7$, $\mathbf{M}_8$ from day two), we need to adjust for the day effect on those chips first. From Yiqiang Luo's MSc thesis (Luo, 2007), we know that day effect does exist using cluster analysis and SAM analysis. Since we are not interested in evaluating the day effect in this report, we simply adjust the day effect by adding difference of mean values between two days to the lower day's expression values for each gene. After the above steps, we obtain a $12,488 \times 8$ matrix of expression values $\mathbf{Y}$ as our data shown in Table 3.1.

We then apply Efron's idea to detect differential expression. Let $\mathbf{Y}_j$ indicate the j-th column of $\mathbf{Y}$, a $12,488$ vector, then $\mathbf{Y}_1$, $\mathbf{Y}_2$, $\mathbf{Y}_3$, and $\mathbf{Y}_4$ come from NOD mice and $\mathbf{Y}_5$, $\mathbf{Y}_6$, $\mathbf{Y}_7$, and $\mathbf{Y}_8$ come from NOD.NOR-Idd4 mice. Let $\bar{Y}_{i(1)}$ and $\bar{Y}_{i(2)}$ be the sample means for gene i under the two conditions (NOD or NOD.NOR-Idd4 mice), then we can get observed expression scores Z_i by Equation (2.2.10). We can also get "null" expression score z_i by permutations, which approximate the unaffected gene samples. The permutation can be done as the following: Represent the original

label for each row as (0 0 0 0 1 1 1 1) (where 0 denotes NOD mice from group I and 1 denotes NOD.NOR-Idd4 mice from group II), then randomly assign four samples for label 0, and the other four samples for label 1. For example, we can set the label pool as (0 1 1 0 1 1 0 0) or (0 1 0 1 1 1 0 0) etc. The total possible number of permutations is $B_{total} = \binom{8}{4} = 70$ here. Next we calculate the corresponding permutated $\bar{Y}^p_{i(1)}$ and $\bar{Y}^p_{i(2)}$ to obtain the test statistic z_i for each gene by Equation (2.2.10), where $\bar{Y}^p_{i(1)}$ and $\bar{Y}^p_{i(2)}$ represent the sample means for gene i under two conditions (0 group or 1 group) corresponding to permutation matrix. Finally, based on the observed expression scores Z_i and the "null" expression scores z_i, after B permutations, we can estimate the relative density ratio $f_0(Z)/f(Z)$ through their relative frequencies by methodology introduced in Section 2.2.1.

3.2 Results of EBarrays in Microarrays – Parametric Empirical Bayes

In previous chapters, we discussed the methodology of parametric empirical Bayes. The methodology is implemented in the **R** package **EBarrays**, which was written by Kendziorski et al. (2003). By **EBarrays**, we can calculate posterior probabilities of patterns of differential expression across multiple conditions.

We will consider two particular specifications of the general mixture models– Gamma Gamma (GG) model and Lognormal Normal (LNN) model. Table 3.2 shows the number of significant genes detected by GG and LNN models when we set the posterior probability p to be 0.5, 0.62 and 0.95 respectively. We also obtain the joint

Posterior probability	$p > 0.5$	$p > 0.62$	$p > 0.95$
GG Model	336	312	232
LNN Model	335	307	225
GG $\bigcap$ LNN	325	304	219

Table 3.2: *Number of significant genes by* ***EBarrays****.*

number of significant genes detected by both models. We can see that there is no big difference of gene numbers detected by these two models. By list of gene names from those two models, the genes detected by those two models are also quite similar. From the above results of **EBarrays**, GG and LNN models of parametric Bayes seem to work very well. But if we take a close look to them, we can find some problems.

The GG and LNN models both involve the assumption of a constant coefficient of variation (CV) and this property is often observed in microarray data (Newton & Kendziorski, 2003). Now we can check this property in our data by **EBarrays**. Ideally, we want to get a constant CV from the data, but from Figure 3.1 we can see that the coefficient of variation does not approximate a constant very well, especially at the right end. So the goodness of fit of parametric empirical Bayes models in our data becomes suspect.

To test the goodness of fit, we then check the Quantile-Quantile (QQ)-plots, marginal density plots for GG model and LNN model. For LNN model, from marginal density plot of Figure 3.2, our data fits quite well except that the empirical kernel density of log expression is a little sharper than the fitted model's distribution in the middle.

The QQ-plot of LNN model in Figure 3.2 shows that, although most of data

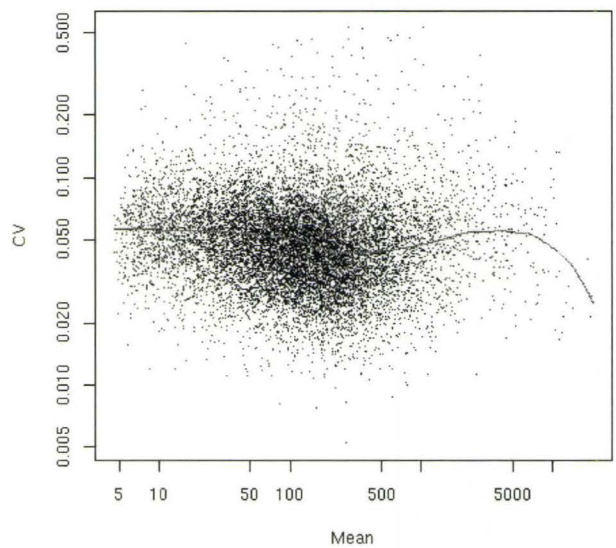

Figure 3.1: *Coefficient of variation (CV) as a function of the mean.*

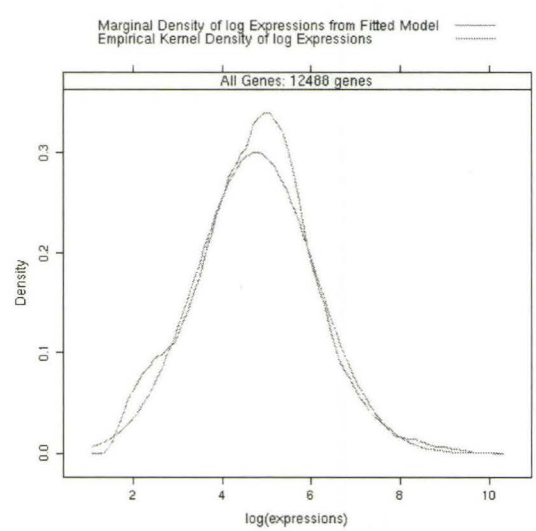

Figure 3.2: *Marginal Densities for Lognormal-Normal Model.*

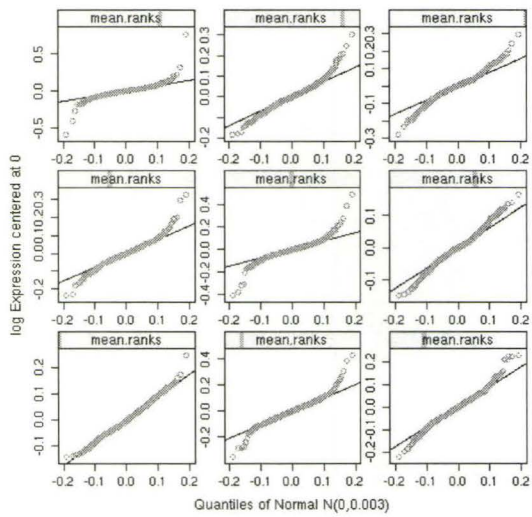

Figure 3.3: *QQ plot for Lognormal-normal model.*

has a straight QQ-plot as we wanted, there are still a number of points departed from the straight line. So we cannot say that our data fit well of the LNN model.

For the GG model, the empirical and theoretical marginal densities of log expressions in Figure 3.4 show the empirical marginal densities do not overlap the theoretical marginal densities, there is a big difference between them. Also, the QQ-plot of GG model in Figure 3.5 shows a big amount of data points depart from the straight line, which means our data do not follow the gamma distribution very well. Therefore, GG model of parametric empirical Bayes is not an appropriate way to analyze our microarray data.

If we take an even closer look at the differentially expressed gene name list, we can find the most serious problem of **EBarrays** for our data. Table 3.3 lists six gene names which appear on the differentially expressed genes' list. Based on biology experimental knowledge, they should not appear there, because they are designed

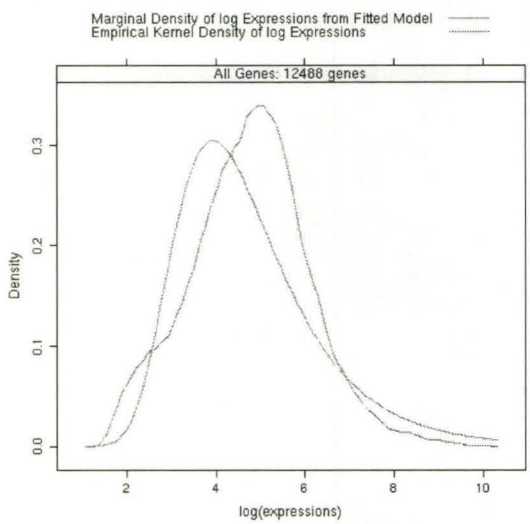

Figure 3.4: *Marginal densities for Gamma-Gamma model.*

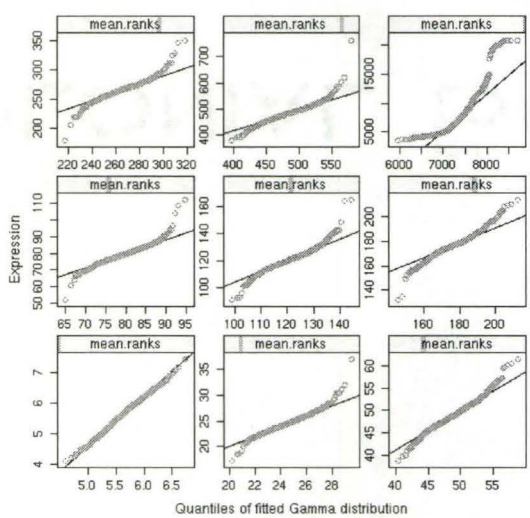

Figure 3.5: *QQ plot for Gamma-Gamma model.*

"AFFX-18SRNAMuR/X00686-M-at"	"AFFX-BioB-3-st"	"AFFX-BioB-5-st"
"AFFX-BioB-M-st"	"AFFX-BioC-5-st"	"AFFX-MURINE-b1-at"

Table 3.3: *Quality control gene names in significant gene name list by* ***EBarrays****.*

for quality control purpose in microarray experiments, they should have unchanged expressions respectively across all chips. So far, why those six genes were detected as differentially expressed is unknown.

Based on the above problems we found by **EBarrays**, we conclude that: we cannot trust LNN and GG models of parametric empirical Bayes methodology in our real Gene Chip data analysis. We will discuss results of nonparametric empirical Bayes in the following sections.

3.3 Results of EBAM in Microarrays – Nonparametric Empirical Bayes

We discussed nonparametric empirical Bayes methodology by Efron et al. (2001) in Section 2.2, this methodology is also implemented in the **R** package **Siggenes** (Schwender et al., 2006), which was written by Holger Schwender. By **Siggenes**, we can calculate posterior probabilities for high-dimensional data to detect differential expression across multiple conditions. Although **Siggenes** works very well on one real microarray data set and some other simulated data (Golub et al., 1999), it shows some serious problems when we apply it to our microarray data.

One key step of Efron's nonparametric empirical Bayes is to get the null score $\{z_i\}$ by generating $B = 20$ independent row-wise sign permutations of $\mathbf{z}$. Ideally, we

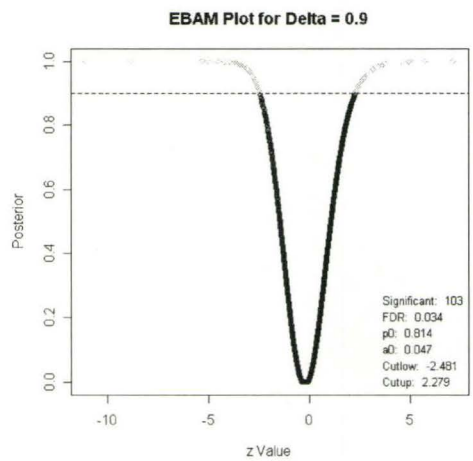

Figure 3.6: *Ideal plot of posterior Vs. Z value.*

would like to get a plot of posterior probability against observed Z value as shown in Figure 3.6. When $|Z|$ value is small, posterior probability of differential expression would approach to 0, because small $|Z|$ value represents little/no difference of expression values between group I genes and group II genes. When $|Z|$ value is bigger, posterior probability of differential expression would approach to 1, because bigger $|Z|$ value represents bigger difference of expression values between group I genes and group II genes.

But when we process this step using a different random start number for function **EBAM** in **Siggenes**, we can get very different results. For example, when we set a random start number $r = 476$, with a permutation matrix as shown in Figure 3.7, we do not get a good solution by **EBAM** as Figure 3.8 uncovered. When we take a closer look to this problem, we found that whenever an extreme pattern such as (0 0 0 0 1 1 1 1) or (1 1 1 1 0 0 0 0) appears in the permutation matrix, **EBAM** failed to

```
> r=476
> perm
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
 [1,]    0    1    1    0    0    1    1    0
 [2,]    1    0    1    0    0    0    1    1
 [3,]    0    1    0    1    0    0    1    1
 [4,]    1    0    0    0    0    1    1    1
 [5,]    0    1    0    1    1    0    0    1
 [6,]    0    0    1    0    0    1    1    1
 [7,]    0    0    1    1    1    0    1    0
 [8,]    1    1    0    0    1    1    0    0
 [9,]    0    0    1    1    0    1    1    0
[10,]    0    1    1    1    0    0    1    0
[11,]    0    1    1    0    0    1    0    1
[12,]    1    0    1    0    1    0    0    1
[13,]    1    0    0    1    0    0    1    1
[14,]    1    0    0    0    1    0    1    1
[15,]    0    0    1    1    0    1    0    1
[16,]    1    0    0    1    1    0    0    1
[17,]    0    1    0    1    0    1    1    0
[18,]    1    1    0    1    0    0    0    1
[19,]    0    0    0    0    1    1    1    1
[20,]    0    1    0    1    0    1    0    1
```

Figure 3.7: *Permutation matrix for* ***EBAM*** *with random seed for permutations* r=476.

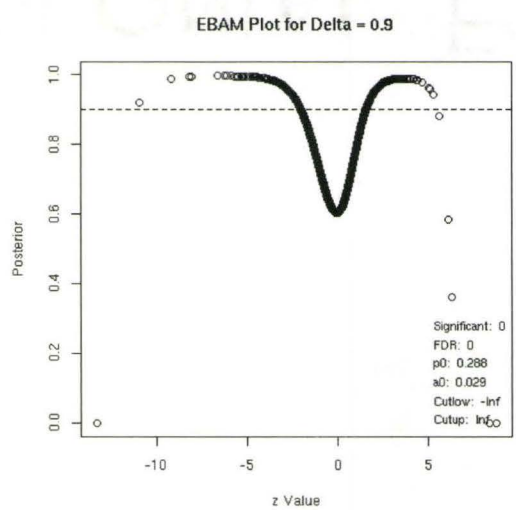

Figure 3.8: ***EBAM*** *plot with random seed for permutations* r=476.

```
> r=321
> perm
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
 [1,]    1    1    1    0    0    0    1    0
 [2,]    1    1    0    1    1    0    0    0
 [3,]    0    1    0    0    1    0    1    1
 [4,]    0    1    0    0    1    1    0    1
 [5,]    0    1    1    0    0    0    1    1
 [6,]    0    1    0    1    1    0    0    1
 [7,]    0    1    1    0    1    0    0    1
 [8,]    0    1    0    0    1    1    1    0
 [9,]    0    1    1    0    0    1    1    0
[10,]    1    0    1    0    1    0    1    0
[11,]    1    0    0    0    1    0    1    1
[12,]    0    1    0    1    0    1    1    0
[13,]    1    0    0    1    1    1    0    0
[14,]    0    0    0    1    1    0    1    1
[15,]    0    1    1    1    0    1    0    0
[16,]    0    0    1    1    0    1    1    0
[17,]    0    1    1    1    1    0    0    0
[18,]    1    1    0    0    1    0    0    1
[19,]    0    1    0    0    0    1    1    1
[20,]    0    1    1    1    0    0    1    0
```

Figure 3.9: *Permutation matrix for* ***EBAM*** *with random seed for permutations* r=321.

detect any differential expression. In the above example, an extreme pattern (0 0 0 0 1 1 1 1) does appear in row 19 of the permutation matrix (see Figure 3.7). Therefore, we conclude that the **EBAM** function cannot detect differential expression when an extreme pattern appears in the permutation matrix.

Furthermore, even when no extreme pattern shows up, **EBAM** can very easily fail in detecting correct differential expressions. For example, with a permutation matrix as shown in Figure 3.9, **EBAM** plot in Figure 3.10 shows the positive end of the posterior curve falls down. Based on the fact that when $|Z|$ increases, posterior probability should monotonically increase at the positive end. Therefore, we can not trust **EBAM**'s result when we set a random start seed for permutations $r = 321$.

Because theoretically we can set any random start number of **EBAM** to get the permutation matrix, based on above problems we found in **Siggenes**, we

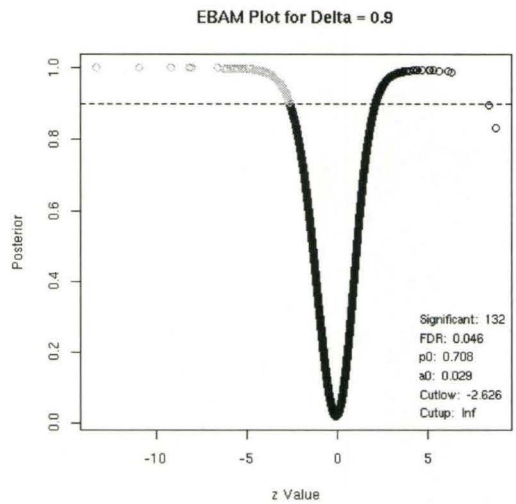

Figure 3.10: ***EBAM*** *Plot with random seed for permutations* r=321.

cannot trust **EBAM** function as nonparametric empirical Bayes to detect differential expression for our microarray data.

3.4 Our Methodology of Nonparametric Empirical Bayes – EBayes

Due to problems we found in **Siggenes**, we apply our own nonparametric empirical Bayes function **EBayes** in **R** based on the idea of Efron et al. (2001). The core code was originally written by Dr. Angelo Canty. In this project, we make several modifications to improve its performance. The final version of the code is presented in Appendix A.1. The key step of nonparametric empirical Bayes is to estimate the relative density of $f_0(Z)/f_1(Z)$.

EBayes attempts to find the posterior probability that each gene is differentially expressed. From Equation (2.2.14), the posterior empirical Bayes probability is obtained by estimating the density ratio $\pi(Z) = \frac{f(Z)}{f(Z)+Bf_0(Z)}$, defined in Equation (2.2.13). We consider three different ideas to implement the estimation of $\pi(Z)$.

- METHOD 1. Logistic regression on quantile interval points.

  We divide the range of observed statistics and permuted statistics into N intervals on the quantile scale. The frequencies of the observed and permuted values in each interval are found and these values are passed to a logistic regression against the interval midpoints with a natural spline on 5 degrees of freedom as the regression function.

- METHOD 2. Smoothing spline fitting on quantile interval points.

  By using the same intervals as Method 1, we find the ratios of frequencies of observed and total frequencies of observed and permuted statistics. A smoothing spline with 5 degrees of freedom is fitted to the logits of the ratios against the interval midpoints. Then the $\pi(Z)$ is the predicted value found from the spline for each observed statistic value.

- METHOD 3. Logistic regression on all data points.

  We create a vector of all observed and permuted values first. Then construct a corresponding vector which is TRUE for each observed value and FALSE for each permuted value. These are then passed as the predictor and response variables to a logistic regression with natural spline regression function with 5 degrees of freedom. Then $\pi(Z)$ is the predicted values at the observed statistics.

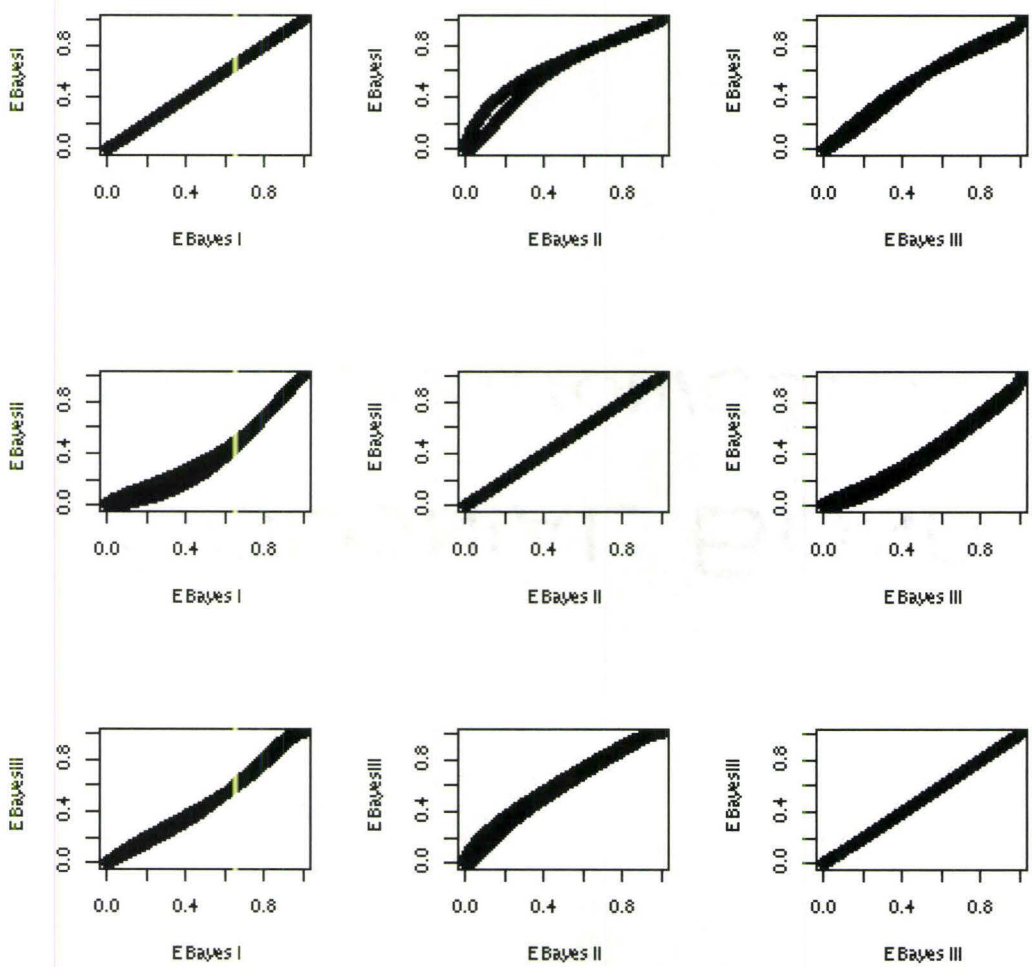

Figure 3.11: *Correlations plot of posterior probabilities of differential expressions among* 3 *methods in* ***EBayes***.

These three methods give some differences in the posterior probabilities. The first and third method are quite similar on average having correlation of 0.995 whereas the second tends to be somewhat more conservative and has correlation of 0.964 and 0.985 with method 1 and method 3 respectively. Figure 3.11 shows the correlation among the three methods in **EBayes**.

Method 1 and method 2 are quite fast in computation time. Method 3 is very computationally intensive compared to the others as it requires passing two vectors of length $(B + 1) \times G$ to the glm function, where B is the number of permuations and G is the number of genes. The results from the third method are closest to that of the **EBAM** method.

3.5 Comparison of EBAM and EBayes in Microarray Data

We test the performance of this new nonparametric empirical Bayes methodology (**EBayes**) on our *MGU74aV2* Affymetrix Gene Chip data. With any random start number and permutation matrix, **EBayes** performs very well. We also compare the results of **EBAM** and **EBayes** under the same conditions (same fudge factor s_0 and the same permutation matrix).

Table 3.4 presents the comparison results of applying the **EBAM** and **EBayes** function to our real microarray data. For each method, the significant gene number (R), estimated False Discovery Rate (FDR), the probability of unaffected genes (p_0), lower cutoff level (Cutlow), upper cutoff level (Cutup) and the method status (Status)

Random No.	Method	R	FDR	p_0	Cutlow	Cutup	Status
r=476	EBAM	0	0.288	0.029	-inf	inf	Problem
r=476	EBayesI	368	0.057	0.592	-2.28	2.854	Normal
r=476	EBayesII	107	0.056	0.811	-2.294	3.931	Normal
r=476	EBayesIII	391	0.059	0.576	-2.221	2.789	Normal
r=321	EBAM	132	0.046	0.708	-2.626	inf	Problem
r=321	EBayesI	340	0.038	0.625	-2.248	3.391	Normal
r=321	EBayesII	165	0.029	0.782	-2.677	3.931	Normal
r=321	EBayesIII	462	0.049	0.658	-2.15	2.642	Normal

Table 3.4: *Comparison of the **EBAM** and **EBayes** procedures to the real microarray data set when setting random start number $r = 476$ and $r = 321$.*

are presented in the table. From the table, we can see that when **EBAM** function fail to detect differential expression, three methods in **EBayes** all work normally.

Figure 3.12 and Figure 3.13 compare the results of **EBayes** and **EBAM** under the same conditions. In Figure 3.12, we can see that **EBAM** fails to detect differential expressions; while in **EBayes**, the posterior probabilities of differential expression approach to 0 when $|Z|$ is close to 0; the posterior probabilities of differential expression approach to 1 when $|Z|$ is far away from 0; and no points fall down on the extremes of x-axis. In Figure 3.13, we can see that no points fall down on the ends of x-axis in **EBayes** plots but two points fall down on the positive end of **EBAM** plot. Those two figures show that when **EBAM** failed to get the differentially expressed genes, all three methods in **EBayes** perform much better when dealing with our real microarray data.

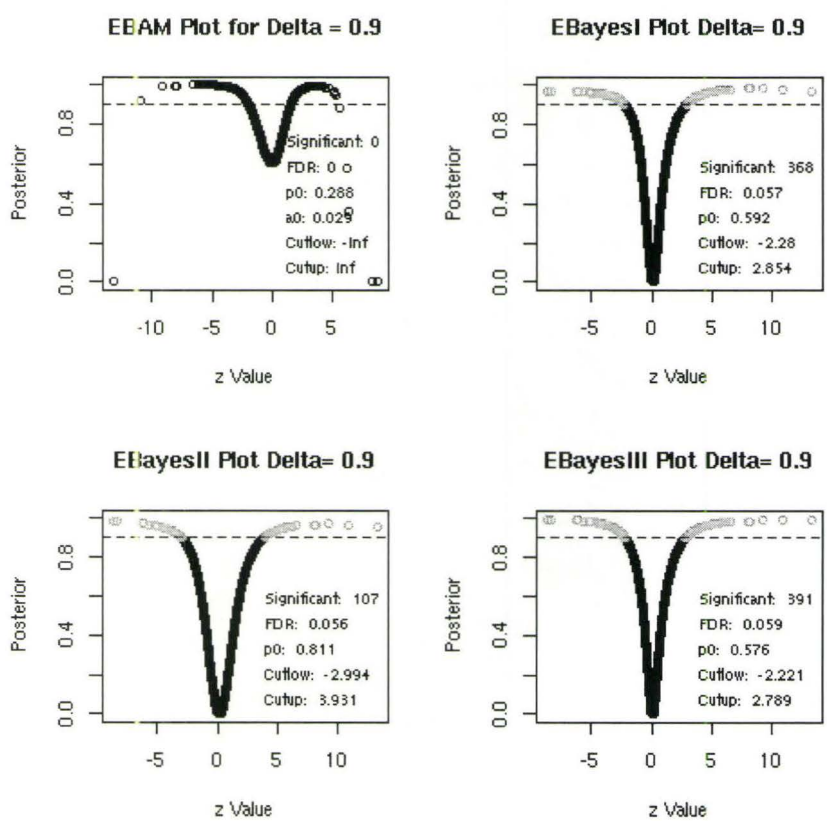

Figure 3.12: *Comparison results between* ***EBAM*** *and* ***EBayes*** *with random seed for permutations* r=476.

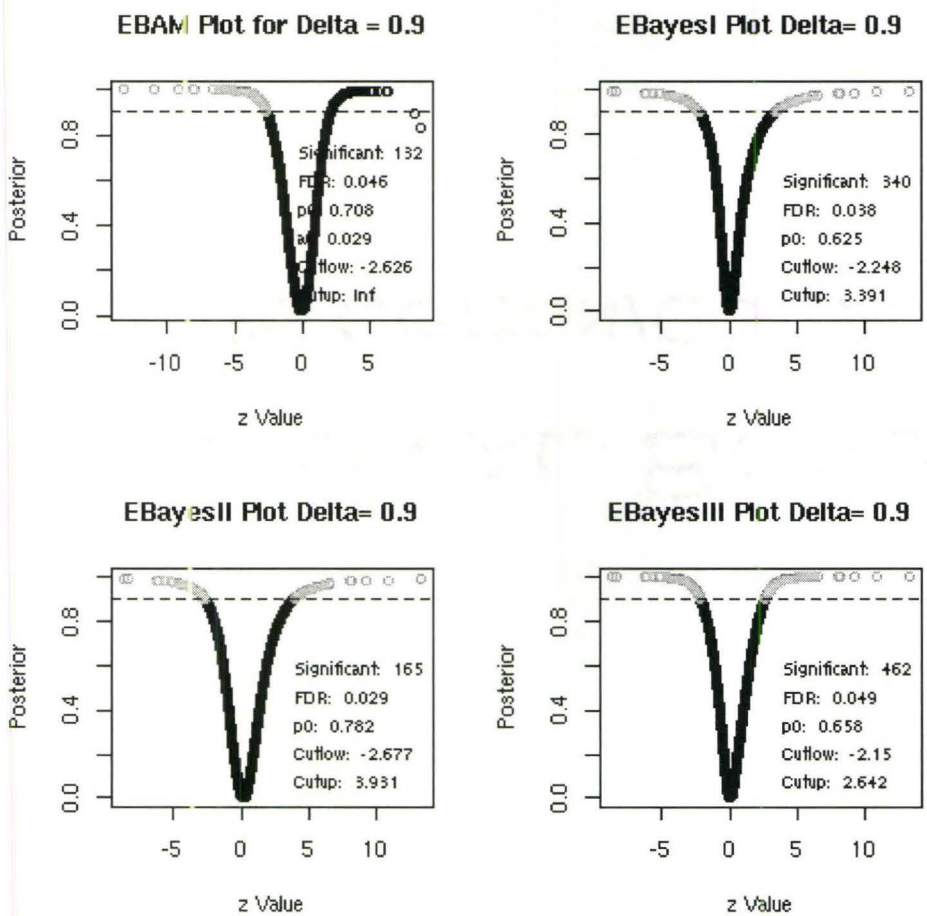

Figure 3.13: *Comparison results between* ***EBAM*** *and* ***EBayes*** *with random seed for permutations r*=321.

Chapter 4

Simulation Results of EBAM Analysis Vs. EBayes Analysis

In last chapter, we can see that **EBayes** analysis works better than the **EBAM** function to our real microarray data because when **EBAM** sometimes fail to detect differential expression, all three methods in **EBayes** function work normally. Whether the overall performance of **EBayes** would be better than **EBAM** needs to be verified by simulations and further research. In simulation study, we know which gene is really differentially expressed by data generating function, so we can compare the estimated FDR and real FDR to see which method can get the most accurate results. Also, we wish to mimic the real microarrays by simulated data to know the performance of the new **EBayes** function to these simulated data. But the closest similarity between the real microarray data and the simulated data cannot be guaranteed due to people's limited knowledge on real human being's genome.

In this chapter, the performance of **EBAM** in **Siggenes** and three nonpara-

metric empirical Bayes approaches in **EBayes** are compared by applying them to three different simulated data sets. In each simulation, $B = 100$ permutations are used to assess the null distribution (i.e., gene is not differently expressed).

4.1 Data Sets and Simulation Procedure

The simulation is performed as follows:

- STEP1. Generate three different simulated data sets.

  Generate a $10,000 \times 10$ matrix μ containing random values drawn from standard normal distributions. Compute the expression level y_{ij} of the i-th gene, $i = 1, \ldots, 10,000$ and the j-th sample, $j = 1, \ldots, n_r \times 2$ by

$$y_{ij} = \mu_{ij} + \begin{cases} \delta_{ij}, & \text{if } i \leq 500 \text{ and } j \leq n_r; \\ \theta_{ij}, & \text{if } 501 \leq i \leq 1000 \text{ and } j \leq n_r; \\ 0, & \text{otherwise.} \end{cases} \tag{4.1.1}$$

  where $\mu_{ij} \sim N(\mu = 5, \sigma = 2)$, $\delta_{ij} \sim \text{seq}(-3, -1, \text{length} = 500) \times \sigma_i$, $\theta_{ij} \sim \text{seq}(1, 3, \text{length} = 500) \times \sigma_i$. Also suppose that the first n_r columns/samples belong to group 1, and the remaining n_r samples belong to group 2. Thus, a data matrix $\mathbf{Y}$ is constructed that contains expression levels of $2n_r$ samples, n_r from each group. The total number of genes is $10,000$ of which the first 10% are differentially expressed.

  For simulation I, set $n_r = 5$; for simulation II, set $n_r = 10$; for simulation III, set $n_r = 20$. i.e., the number of samples from each group is 5, 10 and 20 respectively

for each simulation. Therefore, these three simulations are performed quite similarly except we increase the sample size from 10 in simulation I to 40 in simulation III.

- STEP2. Apply procedures **EBAM** and **EBayes** to above data sets, and record the numbers of differentially expressed genes and the estimated FDRs ($\widehat{\text{FDRs}}$) obtained by these methods. Calculate real FDRs corresponding to these four methods also.

- STEP3. Repeat $M = 100$ times of step 1 and step 2. For each procedure, compute mean numbers of differentially expressed genes and mean values of $\widehat{\text{FDRs}}$ and real FDRs by averaging over iterations. Standard deviations of these quantities have also been calculated.

4.2 Problems of EBayes in Simulation and Modifications

Based on above simulated data sets, we apply each procedure from **EBAM** and **EBayes** to them respectively. For simulation I, **EBAM** and **EBayes** both work well. But when we increase the sample size to 20 (simulation II) and 40 (simulation III), one problem shows up. Ideally, "posterior Vs. Z value" plot should have a "U" shape (as in Figure 3.6) regardless of sample size. But for method I and method III in **EBayes**, we found that posterior probability curve has a "W" shape in the middle when sample size increases, especially for **EBayes**III. The bigger the sample size, the more likely this "W" shape will show up. Figure 4.1 illustrates the problem we found

in one data set of simulation II.

If we check the source code and posterior probability plot from **Siggenes**, we can see that **EBAM** simply replaces the middle part of posterior curve by a horizontal line, which obviously is not appropriate. After different trials, we found that this problem can be solved by decreasing the degree of freedoms of logistic regression in **EBayes** from 5 to 3. Figure 4.2 shows the improvement we have after setting $df = 3$ for **EBayes** in the same data set from simulation II.

By comparing outputs of FDR, significant number of genes, lower cutoff level and upper cutoff level in Figure 4.1 and Figure 4.2, we can see that there are no big differences of theses values in two figures for the same data set. Under different degrees of freedom, these two outputs are almost identical, but the shape of posterior probability curve improved dramatically in the latter figure for **EBayes**I and **EBayes**III. Therefore, the problem of "W" shape posterior probability curve of **EBayes**I and **EBayes**III can be solved by decreasing degrees of freedom in logistic regression of **EBayes** to 3 for bigger size samples.

For **EBayes**II, since so far we have not found "W" shape posterior probability curve, and in terms of power to detect differential expression under the same FDR, setting $df = 5$ can give us a better result. So we will keep $df = 5$ for **EBayes**II. In the next section, we will discuss the comparison results of **EBAM** and **EBayes** when setting $df = c(3, 5, 3)$ for three methods in **EBayes**.

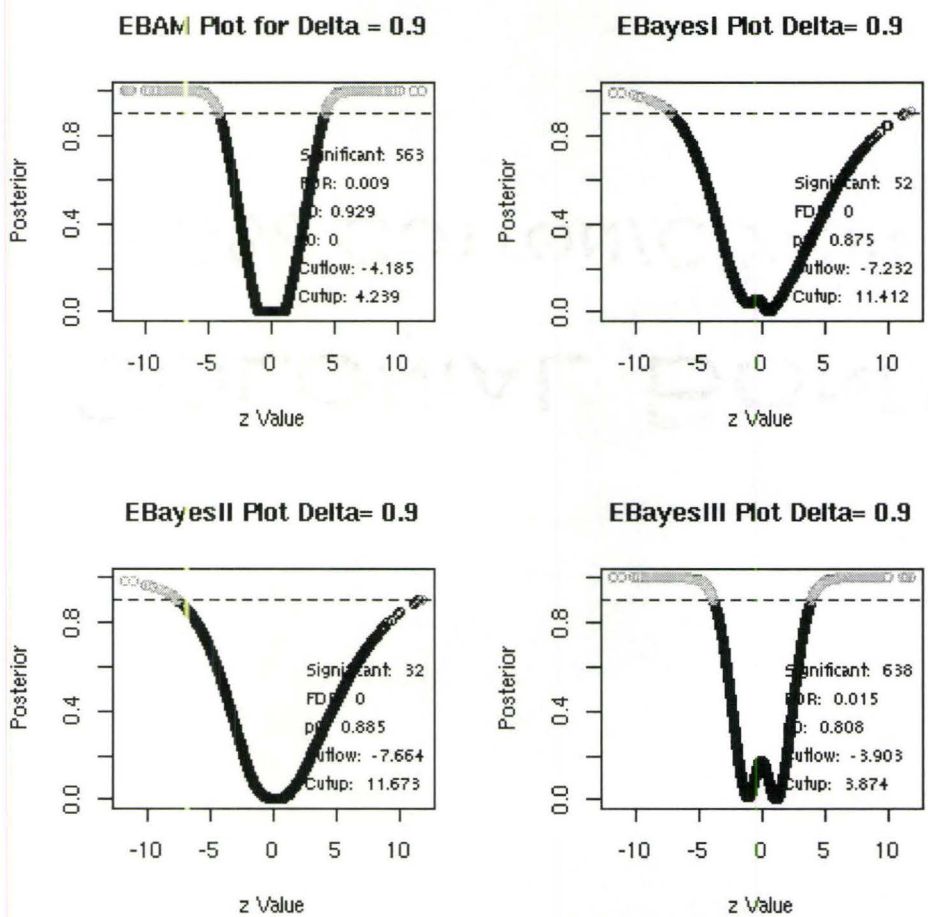

Figure 4.1: *Plot of "posterior Vs. Z value" for simulated data II when set df* $= 5$. *Real FDR for above four methods (clockwise) are: 0.011, 0.000, 0.000, 0.020.*

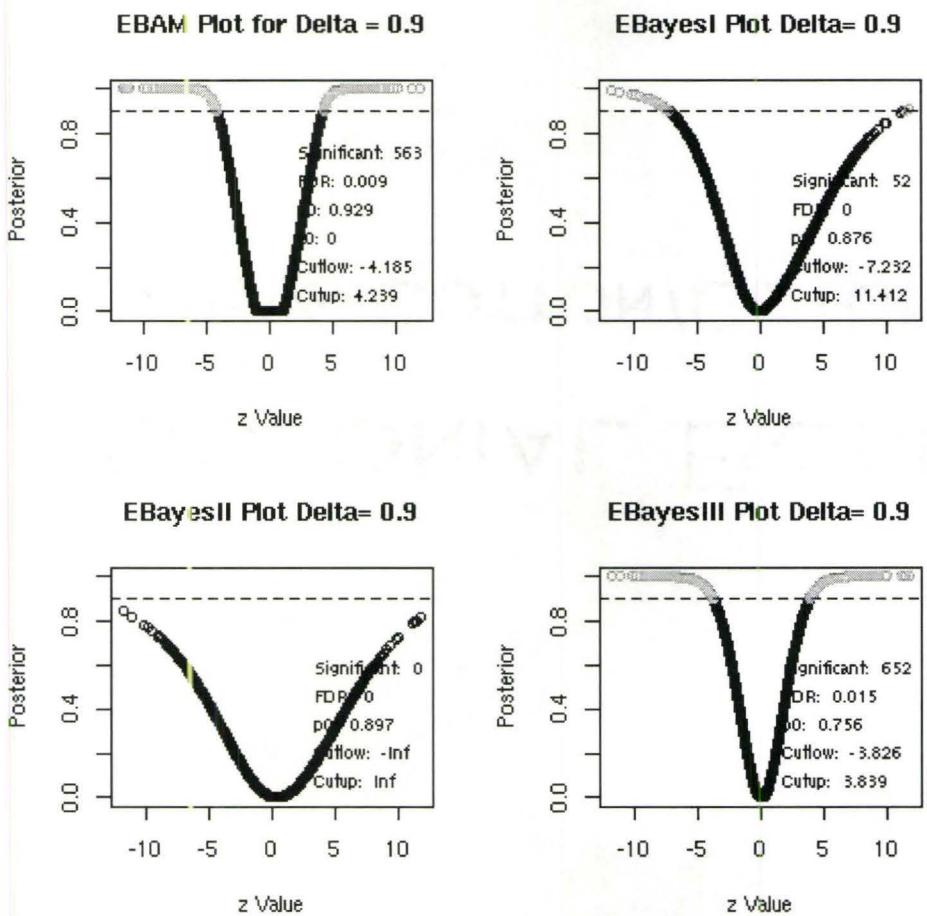

Figure 4.2: *Plot of "posterior Vs. Z value" for simulated data II when set* $df = 3$. *Real FDR for above four methods (clockwise) are: 0.011, 0.000, 0.000, 0.021.*

4.3 Results

In the following, simulation results of the **EBAM** and **EBayes** are summarized in Table 4.1. For each method and data set, the number of identified genes (R), the estimated false discovery rate ($\widehat{FDR}$), and real FDR are listed, where $\widehat{FDR}$ has been defined in Section 2.2.2 as $\widehat{\text{FDR}}_C = \hat{p}_0 \frac{l_p(C)/B}{l_o(C)}$.

For simulation data sets, since we know that the first 10% genes (with row I.D. $i = 1, 2, \cdots, 1000.$) are differentially expressed, then we know exactly which gene is really differentially expressed and which one is not. Thus, we can calculate real FDR by dividing the wrongly claimed significant gene numbers (V) by the number (R) of genes which we claimed significant by above four nonparametric empirical Bayes methodologies. i.e., real FDR=$\frac{V}{R}$. When the significant gene number $R = 0$, we define FDR=0.

From Table 4.1, we can see that when sample size increases, the number of identified genes (R) increases while $\widehat{FDR}$ and real FDR are monotonically decreasing. Which is true since bigger sample size will offer us more information about genes.

For all three simulated data sets, in terms of the number of identified genes (R), $\widehat{\text{FDR}}$ and real FDR, **EBayes**III is quite similar to **EBAM**, and **EBayes**I is similar to **EBayes**II. Under same level of $\widehat{\text{FDR}}$ and real FDR, **EBayes**III and **EBAM** can find approximately same number of differentially expressed genes, so do **EBayes**I and **EBayes**II. But **EBayes**I and **EBayes**II have less power than **EBayes**III and **EBAM** because they detect far fewer significant genes for the same data set under the same cutoff value 0.9. When real differentially expressed gene number is 1000, they can only detect 3 or 4 genes in simulation I, which is too conservative to meet

Data	Simulation I (nr=5)			Simulation II (nr=10)			Simulation III (nr=20)		
Method	R	$\widehat{\text{FDR}}$	FDR	R	$\widehat{\text{FDR}}$	FDR	R	$\widehat{\text{FDR}}$	FDR
EBAM	138	0.0366	0.0294	583	0.0093	0.0097	840	0.0022	0.0023
(s.d.)	21	0.0055	0.0157	17	0.0007	0.0043	10	0.0002	0.0017
EBayes I	4	0.0107	0.0092	56	0.0001	0	163	0	0
(s.d.)	3	0.0088	0.0579	21	0.0001	0	29	0.000005	0
EBayes II	3	0.0088	0.005	41	0.0001	0	137	0	0
(s.d.)	2	0.0090	0.05	15	0.0001	0	24	0.000006	0
EBayes III	133	0.0336	0.0290	663	0.0151	0.0185	890	0.0055	0.0070
(s.d.)	26	0.0039	0.0162	18	0.0006	0.0062	10	0.0004	0.0027

Table 4.1: *Comparison of the* ***EBAM*** *and* ***EBayes*** *procedures to three simulated data sets when setting cutoff level at* 0.9.

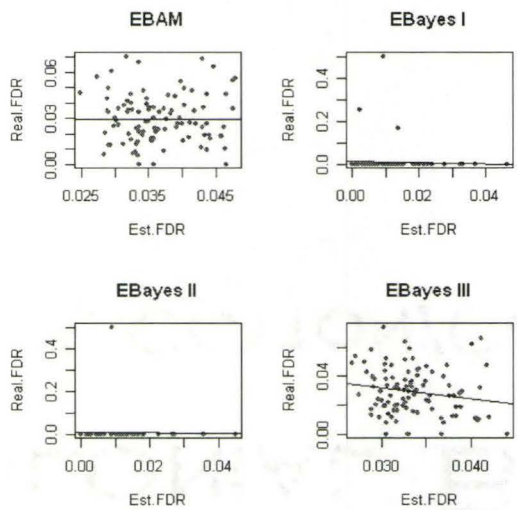

Figure 4.3: *Plot of "Real FDR Vs. Estimated FDR" for simulation I.*

biologists' needs.

Next, we will take a closer look of simulation results for each individual simulated data set.

4.3.1 Simulation I

Figure 4.3 shows relationships between real FDR and estimated $\widehat{FDR}$ among above four methods for simulation I. From this figure, we can see that **EBAM** and **EBayes**III have similar pattern, estimated $\widehat{FDR}$ are quite close to real FDR. Also, **EBayes**I and **EBayes**II have quite similar patterns. For **EBayes**I, only 3 points have relative high real FDR; for **EBayes**II, only one point has a relative high real FDR.

Figure 4.4 shows relationships of posterior probability of differentially expressed genes among **EBAM** and three methods in **EBayes** for a representative

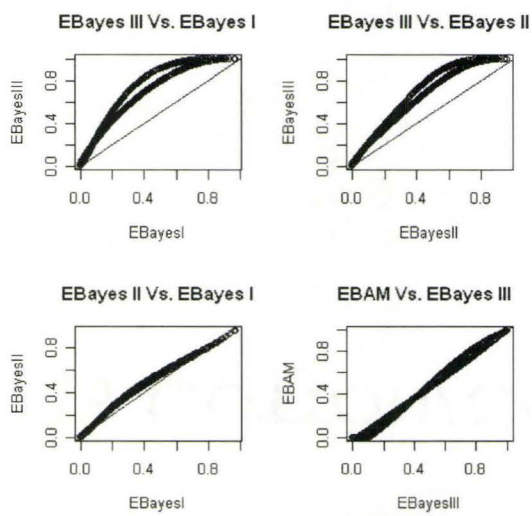

Figure 4.4: *Plots of pairwise relationship of posterior probability for one data set in simulation I.*

data set in simulation I. From this figure, we can see that **EBAM** and **EBayes**III have similar level of posterior probabilities, and which are higher than posterior probabilities of **EBayes**I and **EBayes**II. Higher level of posterior probabilities of **EBAM** and **EBayes**III also help explain why these two methods can detect more differential expressions for the same data set.

4.3.2 Simulation II

Figure 4.5 shows relationships between real FDR and estimated $\widehat{FDR}$ among four methods for simulation II. Similar to simulation I, **EBayes**I and **EBayes**II have similar pattern and **EBAM** and **EBayes**III are similar to each other. Also from the **EBayes**I and **EBayes**II plots, we can see that the total 100 data points all have

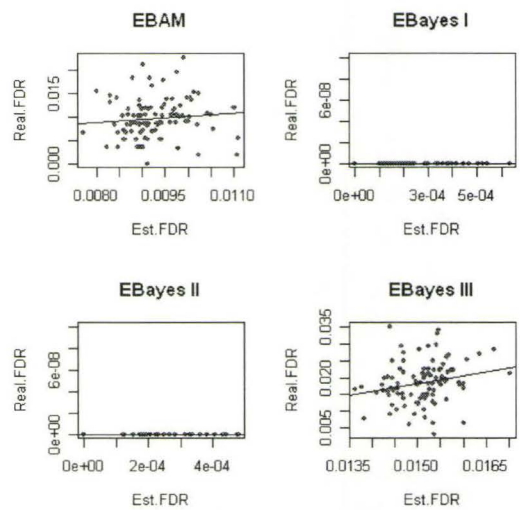

Figure 4.5: *Plot of "Real FDR Vs. Estimated FDR" for simulation II.*

real FDR values at 0, so the regression line in it shrinks to a horizontal line $y = 0$ respectively. This fact tells us that although **EBayesI** and **EBayesII** are conservative to detect differential expressions (they can detect a very small number of significant genes), the genes they claimed significant are very accurate.

Figure 4.6 shows relationships of posterior probability of differentially expressed genes among **EBAM** and three methods in **EBayes** for a representative data set in simulation II. From this figure, we can see that **EBayesI** is highly correlated with **EBayesII**; **EBayesIII** has the highest posterior probability among four methods; while **EBAM** is close to but little less than **EBayesIII**. Thus **EBayesIII** can detect the biggest number of differential expressed genes.

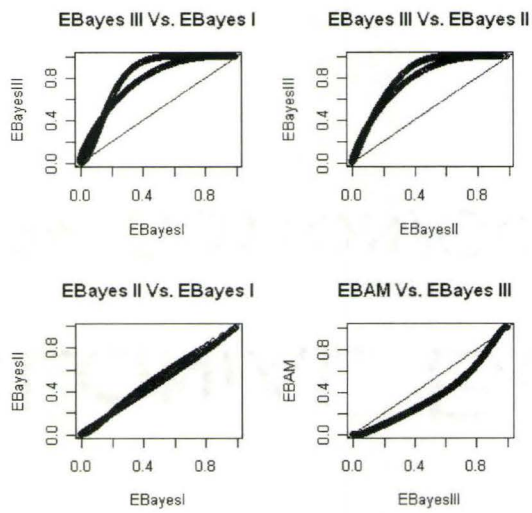

Figure 4.6: *Plots of pairwise relationship of posterior probability for one data set in simulation II.*

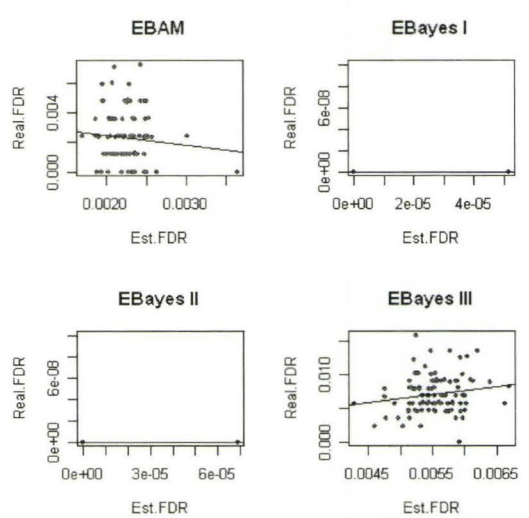

Figure 4.7: *Plot of "Real FDR Vs. Estimated FDR" for simulated data III.*

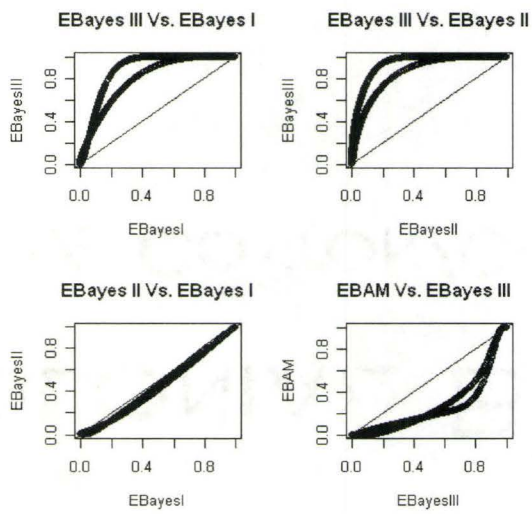

Figure 4.8: *Plot of pairwise relationship of posterior probability for one data set in simulation III.*

4.3.3 Simulation III

Figure 4.7 shows the relationship between real FDR and estimated $\widehat{FDR}$ among above four methods for simulation III. Again, **EBAM** and **EBayes**III have similar patterns and **EBayes**I and **EBayes**II have same patterns. Since the real FDR for all 100 data points are 0 and the estimated $\widehat{FDR}$ are very close to 0 for **EBayes**I and **EBayes**II, the regression line on those plots shrinks to a horizontal line also. The fact that all real FDRs equal to 0 means the significant genes claimed by **EBayes**I and **EBayes**II are very accurate.

Figure 4.8 shows the relationships of posterior probability of differentially expressed genes among **EBAM** and three methods in **EBayes** for a representative data set in simulation III. Again, **EBayes**I is highly correlated with **EBayes**II. **EBayes**III

has the highest posterior probability among four methods, which also explained why **EBayesIII** can detect the biggest number of differential expressed genes for the same data set in simulation III.

In summary, plots of "Real FDR Vs. Estimated FDR" and pairwise relationships of posterior probability among **EBAM** and **EBayes** for three simulations both show **EBayesI** is highly correlated with **EBayesII** and **EBAM** is highly correlated with **EBayesIII**. But **EBayesI** and **EBayesII** can only detect very small number of differentially expressed genes. This will limit their applications to our real microarray data analysis. Furthermore, **EBayesIII** has the highest posterior probability of differential expression among four methods. This also explains why **EBayesIII** can detect more number of differential expressions than **EBayesI** and **EBayesII** under the same cut off level $p = 0.9$.

From Equation (2.2.12), we know that the posterior probability of differential expression $p_1(Z) = 1 - p_0 \frac{f_0(Z)}{f(Z)}$, so we may guess that **EBayesIII** has the highest $p_1(Z)$ due to lower value of p_0 among four methods. But after a close check on values of p_0, we cannot see this relationship. Therefore, we know **EBayesIII** can detect more differentially expressed genes because it can get a better estimated ratio of $\frac{f_0(Z)}{f(Z)}$. Figure 4.9 illustrated the relationship of p_0 among four methods in simulation III. We can get similar patterns in simulation I and simulation II.

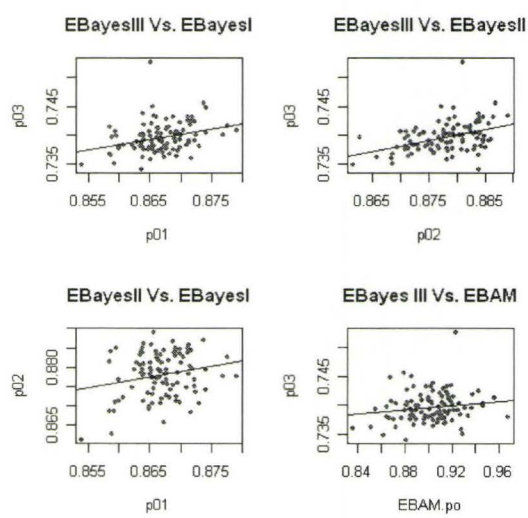

Figure 4.9: *Plots of pairwise relationships of p_0 for simulation III.*

Chapter 5

Discussions and Future Work

In this project, we applied parametric and nonparametric empirical Bayes to detect differentially expressed genes in our Affy microarray data set.

Results of **EBarrays** in **R** show that this parametric empirical Bayes does not work well because our microarray data do not follow ideal Lognormal-normal model or Gamma-gamma model, the only models for which parametric empirical Bayes is offered so far. Whether there are any other good parametric models for microarray data analysis may be discussed in the future. Also, why those six weird probe sets (see Table 3.3) originally for quality control purpose appeared in differentially expressed gene list by **EBarrays** could be discussed further.

We also did nonparametric empirical Bayes analysis on microarrays based on Efron's idea. One nonparametric empirical Bayes package called **Siggenes** in **R** does not work well also. We found that with different random start numbers, **EBAM** function in **Siggenes** could fail if an extreme pattern appears in permutation matrix. Furthermore, even if there is no extreme pattern, sometimes the results from **EBAM**

still cannot be trusted. Some points at the extremes of posterior probability curve fall down, but this should not happen according to Efron's methodology.

We applied our own **R** function called **EBayes** to implement Efron's (Efron et al., 2001) nonparametric empirical Bayes methodology. The real microarray data analysis on Affymetrix GeneChip *MGU74aV2* data and some simulated data sets show that **EBayes** works very well, especially when **EBAM** fails to detect differentially expressed genes. Therefore, **EBayes** is a good nonparametric empirical Bayes methodology to analyze microarrays.

The original simulation results show that when sample size increases, a '*W*' shape could appear in the 'Posterior Vs. Z value' plots for **EBayes**I and **EBayes**III. But this problem could be solved by changing the degrees of freedom from $df = 5$ to $df = 3$ in the logistic regression of **EBayes** without making big difference on estimating $\widehat{\text{FDR}}$s and detecting differentially expressed gene numbers (R). We conclude that for bigger simulated microarray data sets (sample size $n \geq 10$), $df = 3$ is the best choice for **EBayes**I and **EBayes**III; and $df = 5$ is the best choice for **EBayes**II. Since we get this conclusion through practical trials, another systematic methodology of finding optimal degrees of freedom for logistic regression in **EBayes** for all different sample size could be discussed in the future.

Three differently sized simulations also show that **EBayes**I is highly correlated with **EBayes**II, and **EBayes**III is highly correlated with **EBAM**. Since the former two methods are too conservative to detect differential expressions, **EBayes**III and **EBAM** have better performance in simulated microarray data sets. Furthermore, with a concern that the 'Posterior Vs. Z value' plots for **EBAM** could easily have some drop off points at the extremes of x-axis in real microarray analysis, we be-

lieve that **EBayes**III is the best methodology in detecting differential expression in microarrays.

Appendix A

R Codes

A.1 R Codes for EBayes

```
EBayes <- function(obs, perms, nint=200, df=c(3,5,3), by.range=F) {

 ##########################################################################
 # Three methods in this function are tried to implement Efron's
 ##    nonparametric empirical Bayes idea to find posterior probability
 ##    of differentially expressed genes. A main difference among
 ##    these three methods is the way they find pi(Z) as Equation
 ##    (2.2.15) defined. Method1 applies logistic regression on
 ##    quantile interval points; Method2 uses smoothing spline fitting
 ##    on quantile interval points; and Method3 applies logistic
 ##    regression on all data points.

 # The following values should be input into EBayes function:
 # obs   = the observed Z values
 # perms = the permutated z values
 # nint  = number of intervals be used to get relative frequencies
 # df    = degrees of freedom of logits function for 3 methods
 # by.range= True/False. If it is True, the whole data range is
 ##   a combination of all points from observed Z values and 2 end
 ##   points from the permutated z values; if it is False, the whole
 ##   data range is a combination of all points from observed Z
 ##   values and permutated z values.

 # The output of this function is a list contains two vectors:
 # pr=c(pr1, pr2, pr3). It is a G*3 matrix which corresponding
```

```
##   to the posterior probabilities of differentially expressed
##   genes calculated by three methods;
# p0=c(p01, p02, p03). It is a 1*3 vector including the
##   probabilities of unaffected genes from EBayes 3 methods.
#############################################################

 if (by.range)
   alldata <- c(obs,range(perms))
 else
   alldata <- c(obs,perms)
 # Method 1.
 breaks <- quantiles(alldata, (0:(nint+1))/(nint+1))
 mids <- (breaks[-(nint+1)]+breaks[-1])/2
 freq.obs <- table(cut(obs, breaks, include.lowest=TRUE))
 freq.perms <- table(cut(perms, breaks, include.lowest=TRUE))
 freqs <- cbind(freq.obs, freq.perms)
 mod <- glm(freqs~ns(mids, df=df[1]), family=binomial(logit))
 piZ <- predict(mod, data.frame(mids=obs), type="response")
 fratio <- (1-piZ)/(ncol(perms)*piZ)
 p01 <- min(c(1, 1/fratio))
 pr1 <- 1-p01*fratio

 # Method 2.
 ratio <- freqs[,1]/(freqs[,2]+freqs[,1])
 mod <- smooth.spline(mids, logit(ratio), df=df[2])
 piZ <- inv.logit(predict(mod, obs)$y)
 fratio <- (1-piZ)/(ncol(perms)*piZ)
 p02 <- min(c(1, 1/fratio))
 pr2 <- 1-p02*fratio

 # Method 3.
 x <- c(obs, perms)
 y <- rep(c(TRUE, FALSE), c(length(obs), length(perms)))
 mod <- glm(y~ns(x, df=df[3]), family=binomial(logit))
 piZ <- predict(mod, data.frame(x=obs), type="response")
 fratio <- (1-piZ)/(ncol(perms)*piZ)
 p03 <- min(c(1, 1/fratio))
 pr3 <- 1-p03*fratio
 list(pr=cbind(pr1, pr2, pr3),p0=cbind(p01,p02,p03))
}
```

A.2 R Codes for FDR Calculation

```
EBayes.FDR <- function(obs,perms,pr=pr,p0=1,delta=0.9) {

 ##################################################################
 # The following values should be input into EBayes.FDR function:
 # obs  = the observed Z values
 # perms= the permutated z values
 # pr   = the posterior probability corresponding to observed Z value
 # p0   = the probabilities of unaffected genes from EBayes function
 # delta= cutoff level of posterior probability of differentially
 ##       expressed genes, it can be any number in (0,1]

 # This function's output is a 5-column-table with values:
 # Delta = cutoff level of posterior probability on differential
 ##       expressions, it can be any number in (0,1]
 # Number= the significant number of differential expressions
 # FDR   = estimated FDR defined by Equation (2.2.19)
 # CL    = lower cutoff level on observed Z values
 # CU    = upper cutoff level on observed Z values
 ##################################################################

  if (any(delta<=0 | delta>1))
    stop("The delta values must be in (0,1]")
  pr <- pr[order(obs)]
  obs <- sort(obs)
  m <- length(obs)
  out <- matrix(0,length(delta),5)
  colnames(out) <- c("Delta","Number","FDR","CL","CU")
  for (i in 1:length(delta)) {
    sig.ids<-which(pr>=delta[i])
    out[i,2]<-length(sig.ids)
    #out[i,3]<-mean(1-pr[sig.ids]) # get local FDR
    neg.ids <- which(obs<0&pr>=delta[i])
    if (length(neg.ids)>0)
      out[i,4]<-obs[max(neg.ids)]
    else
      out[i,4]<- -Inf
    pos.ids <- which(obs>0&pr>=delta[i])
    if (length(pos.ids)>0)
```

```
      out[i,5]<-obs[min(pos.ids)]
    else
      out[i,5]<- Inf
    perms.ids<-which(perms<=out[i,4] | perms>=out[i,5])
    out[i,3]<-p0*length(perms.ids)/ncol(perms)/max(out[i,2],1)
  }
  out
}
```

A.3 R Codes for EBayes Plot

```
EBayes.plot<- function(obs,perms,pr,p0=1,delta=0.9, main="EBayes Plot") {

 ##########################################################################
 # The following values should be input into EBayes.plot function:
 # obs  = the observed Z values
 # perms= the permutated z values
 # pr   = the posterior probability corresponding to observed Z value
 # p0   = the probabilities of unaffected genes from EBayes function
 # delta= cutoff level of posterior probability of differentially
 ##       expressed genes, it can be any number in (0,1]
 # main = the title of the plot

 # This function's output is a plot of 'Posterior against Z value'
 ##    to illustrate the relationship between posterior probabilities
 ##    of differentially expressed genes and observed Z values, which
 ##    are calculated by EBayes function. It has a legend at the
 ##    bottom right with the following values:

 # Significant= the significant number of differential expressions
 # FDR   = estimated FDR defined by Equation (2.2.19)
 # p0    = the probabilities of unaffected genes from EBayes function
 # Cutlow= lower cutoff level on observed Z values
 # Cutup = upper cutoff level on observed Z values

 # The data points which are claimed as significant will be marked
 ##    as green color while the majority of unaffected genes are
 ##    marked as black color. If no points are claimed significant,
 ##    a dashed horizontal line at the value of delta will vanish.
```

```
#####################################################################

  out<-EBayes.FDR(obs,perms,pr,p0,delta)
  ids<-which(obs<=out[1,4] | obs>=out[1,5])
  main<-paste(main, "Delta=", delta[1])
  xlab<-"z Value"
  ylab<-"Posterior"
  if (length(ids)==0)
     plot(obs,pr,main=main,xlab=xlab,ylab=ylab,ylim=c(0,1))
  else {
    plot(obs[-ids],pr[-ids],main=main,xlab=xlab,ylab=ylab,
         xlim=range(obs),ylim=c(0,1))
    points(obs[ids],pr[ids],col=3)
  }
  abline(h=delta,lty="dashed")
  tmp1<-c("Significant:", "FDR:", "p0:","Cutlow:","Cutup:")
  tmp2<-c(out[1,2],round(out[1,3],3),round(p0,3),
          round(out[1,4],3),round(out[1,5],3))
  textlegend<-paste(tmp1,tmp2,sep="    ")
  legend("bottomright",legend=textlegend,cex=0.8,bty="n",y.intersp=1.3)
}
```

A.4 R Codes for Simulation

```
simuyn <- function(G,nr,G0=G*0.1,mu,sd) {

  ###################################################################
  # This function can give us a G*(nr*2) simulated data matrix yn.
  # nr= number of samples for each strain
  # G = total number of genes
  # G0= number of differentially expressed genes
  # mu= mean of the normal distribution
  # sd= standard deviation of normal distribution
  # diseq= difference between differentially expressed genes
  ##       and equally expressed genes

  # For example, we can get a 10000*(nr*2) data matrix yn with row
  ##    names 1,2,...,10000 as the following:
  # nr<-5 # there are 5 samples in each strain
```

```
  # G <-10000
  # G0<-G*0.1
  # mu<-5
  # sd<-2
  ##############################################################

  mui<-rnorm(G,mu,sd)
  sdi<-sqrt(rchisq(G,1));
  diseq<-c(seq(-3,-1,length=G0/2),seq(1,3,length=G0/2))
  diseq<-diseq*sdi[1:G0]
  di<-c(diseq,rep(0,G-G0))

  strain1=matrix(rnorm(G*nr,rep(mui,nr),rep(sdi,nr)),nrow=G,ncol=nr);
  strain2=matrix(rnorm(G*nr,rep(mui+di,nr),rep(sdi,nr)),nrow=G,ncol=nr);
  yn<-cbind(strain1,strain2)
  row.names(yn)<-1:G
  yn
}
```

References

Affymetrix. (1999). *Affymetrix Microarray Suite User Guide.* Affymetrix, Santa Clara.

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*, 289-300.

Boutros, P. C. (2007). Fun with microarrays part ii: Data analysis. *Hypothesis-A Journal for the Discussion of Science*, *5*, 15-22.

Casella, G. (1985). An introduction to empirical Bayes data analysis. *Journal of the American Statistical Association*, *39*, 83-87.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B*, *39*, 1-38.

Dudoit, S., Yang, Y., Callow, M., & Speed, T. (2002). Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Statistica Sinica*, *12*, 111-139.

Efron, B., & Morris, C. (1977). Stein's paradox in statistics. *Scientific American*, *236*, 119-127.

Efron, B., Tibshirani, R., Storey, J. D., & Tusher, V. (2001). Empirical Bayes analysis of a microarray experiment. *Journal of the American Statistical Association*, *96*, 1151-1160.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., et al. (1999). Molecular classfication of cancer: class discovery and class prediction by gene expression monitoring. *Science*, *286*, 531-537.

Irizarry, R. A., Gautier, L., & Cope, L. M. (2003). An R package for analyses of affymetrix oligonucleotide arrays. In G. Parmigiani, E. S. Garrett, R. A. Irizarry, & S. L. Zeger (Eds.), *The Analysis of Gene Expression Data: Methods and Software.* Springer.

Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y., Antonellis, K., Scherf, U., et al. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, *4*, 249-264.

Kendziorski, C. M., Newton, M. A., Lan, H., & Gould, M. N. (2003). On parametric

empirical bayes methods for comparing multiple groups using replicated gene expression profiles. *Statistics in Medicine*, *22*, 3899-3914.

Li, C., & Wong, W. (2001). Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection. *Proceedings of the National Academy of Science*, *98*, 31-36.

Llanos, G., & Libman, I. (1994). Diabetes in the Americas. *Bulletin of the Pan American Health Organization*, *28*, 285-301.

Lockhart, D. J., Dong, H., Byrne, M. C., Follettie, M. T., Gallo, M. V., Chee, M. S., et al. (1996). Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology*, *14*, 1675-1680.

Luo, Y. (2007). *Comparison Between Affymetrix and Illumina Gene Expression Microarray Platforms.* Unpublished master's thesis, McMaster University, Canada.

Newhook, L. A., Curtis, J., Hagerty, D., Grant, M., Paterson, A. D., Crummel, C., et al. (2004). High incidence of childhood type 1 diabetes in the Avalon Peninsula, Newfoundland, Canada. *Diabetes Care*, *27*, 885-888.

Newton, M., & Kendziorski. (2003). Parametric empirical Bayes methods for microarrays. In G. Parmigiani, E. S. Garrett, R. A. Irizarry, & S. L. Zeger (Eds.), *The Analysis of Gene Expression Data: Methods and Software.* Springer.

Robbins, H. (1955). An empirical bayes approach to statistics. In *Proceedings of the Third Berkeley Symposium Mathematical Statistics and Probability 1* (p. 157-164). Berkeley: University of California Press.

Schwender, H., Krause, A., & Ickstadt, K. (2006). Identifying interesting genes with siggenes. *R News*, *6*(5), 45-50.

Storey, J. D. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society, Series B*, *64*, 479-498.

Storey, J. D., & Tibshirani, R. (2001). Estimating false discovery rates under dependence, with applicaions to dna microarrays. *Technical Report 2001-28, Department of Statistics, Stanford University.*.

Tusher, V., Tibshirani, R., & Chu, C. (2001). Significance analysis of microarrays applied to transcriptional responses to ionizing radiation. *Proceedings of the National Academy of Sciences*, *98*, 5116-5121.

Watson, J. D., & Crick, F. H. C. (1953). Genetical implications of the structure of deoxyribonucleic acid. *Nature*, *171*, 964-967.

Yang, R. (2006). *Comparison of Preprocessing Methods for DNA Microarray Data.* Unpublished master's thesis, McMaster University, Canada.