1-1-2009

DETECTING DIFFERENTIALLY EXPRESSED GENES WHILE CONTROLLING THE FALSE DISCOVERY RATE FOR MICROARRAY DATA

SHUO JIAO
University of Nebraska at Lincoln, jiao.shuo@gmail.com

DETECTING DIFFERENTIALLY EXPRESSED GENES WHILE CONTROLLING THE FALSE DISCOVERY RATE FOR MICROARRAY DATA

by

Shuo Jiao

A DISSERTATION

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Doctor of Philosophy

Major: Statistics

Under the Supervision of Professors Shunpu Zhang and Stephen D. Kachman

Lincoln, Nebraska

December, 2009

DETECTING DIFFERENTIALLY EXPRESSED GENES WHILE CONTROLLING THE FALSE DISCOVERY RATE FOR MICROARRAY DATA

Shuo Jiao, Ph.D.

University of Nebraska, 2009

Advisers: Shunpu Zhang and Stephen D. Kachman

Microarray is an important technology which enables people to investigate the expression levels of thousands of genes at the same time. One common goal of microarray data analysis is to detect differentially expressed genes while controlling the false discovery rate. This dissertation consists with four papers written to address this goal. The dissertation is organized as follows: In Chapter 1, a brief introduction of the Affymetrix GeneChip microarray technology is provided. The concept of differentially expressed genes and the definition of the false discovery rate are also introduced. In Chapter 2, a literature review of the related works on this matter is provided. In Chapter 3, a t-mixture model based method is proposed to detect differentially expressed genes. In Chapter 4, a t-mixture model based false discovery rate estimator is proposed to overcome several problems of the current empirical false discovery rate estimators. In Chapter 5, a two-step false discovery rate estimation procedure is proposed to correct the overestimation of the false discovery rate caused by differentially expressed genes. In Chapter 6, a novel estimator is developed to estimate the proportion of equivalently expressed genes, which is an important component of the false discovery rate estimators. In Chapter 7, a summary of the dissertation will be given along with some possible directions for the future work.

Acknowledgements

The completion of this dissertation is impossible without the support from many people. I would like to give my deepest gratitude to my advisor Dr. Shunpu Zhang. He directed me into the area of my dissertation, gave me insightful advices, and encouragingly supported my ideas. I would also like to thank my co-advisor Dr. Stephen D. Kachman for always being there to listen and discuss. I learned a lot from his way of thinking. I would like to thank Dr. Kent M. Eskridge and Dr. Istvan Ladunga for serving on my PhD. supervisory committee. Their careful proofreading of the dissertation proposal helps me improve my writing skills and I am grateful to them for holding me to a high research standard.

I am also indebted to Dr. Walter W. Stroup, Dr. Jim Lewis, and Dr. Ruth Heaton for providing the financial support to me, which was crucial for my PhD program. I want to give a special thanks to Dr. Yuannan Xia for letting me participate in his microarray experiment.

I dedicate this work to my parents, my fiancee, and our family who have been supportive all the time.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

1.1. Background

Gene expression is an important process in molecular biology. When the DNA sequences in a gene are transcribed into mRNA, this gene is said to be ”expressed”, and the concentration of mRNA is called the ”expression level” of this gene. Gene expression profiling has proved to be helpful in many areas, such as understanding the global cellular function and the molecular mechanisms underlying certain biological processes. For example, we know that the growth, division, and death of a cell are all controlled by the genes in the cell. When some genes do not function properly, the cell growth may get out of control, which may lead to cancer. Interestingly, those cancer-related genes often have different expression levels in cancer cells compared to healthy cells. Hence, if we can detect differentially expressed (DE) genes between cancer and healthy cells, those detected genes are associated with the cancer.

Before microarray technology was invented, scientists can only study one or maybe a few genes at a time. Since the number of genes in a living organism is usually huge, it would take a very long time to investigate all of them. Microarray technology makes it possible to monitor the expression levels of thousands of genes simultaneously. Currently, the most commonly used microarray technology is Affymetrix GeneChip System, which usually consists of a microarray chip, a hybridization oven, a fluidics station, a scanner, and a computer workstation. An Affymetrix microarray chip is a microscope slide on which every gene is represented by a probe

set of 10 - 25 oligonucleotide pairs (probe pairs). For every probe pair, there is one oligonucleotide perfectly matching (PM) to the gene sequence and the other oligonucleotide mismatching (MM; same as PM but with a single homomericbase change for the middle base) to the gene sequence. The purpose of including a mismatching probe is to determine the impact of background and nonspecific hybridization. Once the microarray chips have been obtained, the next step is to prepare biotin-labeled RNA samples from the tissue. Then the biotin-labeled RNA is hybridized to the microarray chip in a hybridization oven. After that the hybridized microarray chip will be washed and stained with one kind of fluorescence called phycoerythrin-conjugated streptavidin in the fluidics station. At last, the stained microarray chip will be scanned in the GeneChip scanner.

From the output image of the scanner, the color intensities for both probes (PM and MM) in each probe pair can be obtained. A weighted average of all the probe pair differences (PM-MM) in a probe set will be computed as the signal for that probe set, which is also the signal for the corresponding gene of that probe set. After the signals of all the genes have be obtained, the data will be saved as a *.CEL data file.

The procedure described above is for a single microarray experiment. In practice, we always need to repeat a microarray experiment several times to get valid statistical inferences. For multiple microarray experiments, normalization is necessary to correct the technical or biological variations among different experiments such as cross-hybridization. One widely used normalization method is called Robust Multichip Average (RMA; Irizarry *et al.* (2003)), which consists of three steps: background correction, quantile normalization and expression calculation. After the normalization process, the final microarray data from multiple experiments can

be summarized as an expression level matrix. Every row represents a gene, every column represents an microarray experiment (replicate), and every entry is the corresponding expression level. Due to the expense of a microarray experiment, the number of replicates is typically small.

1.2. Problem Statement

A common goal of analyzing microarray data is to detect genes with differential expression under two conditions. In other words,

Definition 1.1. *Y_{ij} is the expression of gene i in experiment j ($i = 1, 2, ..., n$; $j = 1, ..., j_1$, j_1+1,..., j_1+j_2=J), and the first j_1 and last j_2 experiments are obtained under the two different conditions. $E(Y_{ij}) = \mu_{i1}$ if $j \leq j_1$; $E(Y_{ij}) = \mu_{i2}$ if $j > j_1$.*

For every gene i, the null hypothesis is $H_0 : \mu_{i1} = \mu_{i2}$ and the alternative hypothesis is $H_1 : \mu_{i1} \neq \mu_{i2}$ given Definition 1.1. As we can see, microarray data analysis is basically a multiple hypothesis testing problem with a large number of hypotheses and a small number of replicates. Hence, regular method such as two sample t-test is not appropriate here because of the lack of statistical power.

Controlling family-wise error rate (FWER) with Bonferroni adjustment is a common practice in regular multiple hypothesis testing problems. For example, if the number of genes is n and the genome wide FWER is controlled at significance level α, then the Type I error rate for an individual gene is controlled at α/n.

However, due to the large number of genes in microarray data, the false discovery rate (FDR) introduced by Benjamini and Hochberg (1995) is now commonly used as the choice of control criterion in microarray studies. Suppose Table 1.1 is the outcome of a microarray data

Table 1.1. Outcome of a microarray data analysis with n genes.

	Accept	Reject	Total
Equivalently expressed (EE) genes	TN	FP	n_0
Differentially expressed (DE) genes	FN	TP	$n - n_0$
	N	P	n

analysis, the FDR is defined as

$$FDR = E[\frac{FP}{P}],$$

where FP is the number of rejected EE genes, or false positive genes; and P is the total number of significant genes.

It was shown in Storey and Tibshirani (2003) that the FDR can be approximated by

$$FDR \approx \frac{E[FP]}{E[P]}. \tag{1.1}$$

Notice that the number of false positive genes (FP) is the number of EE genes which are falsely called positive. By definition,

$$FP = n_0 * \text{Type I error rate}, \tag{1.2}$$

where n_0 is the number of EE genes. In practice, both type I error rate and n_0 need to be estimated. As a result, (1.1) can be re-written as:

$$\begin{aligned} FDR \approx & \frac{E[n_0 * \text{Type I error rate}]}{E[P]} \\ = & \frac{(n_0/n)E[n * \text{Type I error rate}]}{E[P]} \\ = & \frac{\pi_0 E[FP^*]}{E[P]}, \end{aligned} \tag{1.3}$$

where FP^* is the number of false positive genes when all genes are EE genes and π_0 is the proportion of EE genes. In (1.3), $E[P]$, $E[FP^*]$ and π_0 all need to be estimated in practice.

It has been proved that in many cases controlling FDR is more appropriate compared to controlling FWER. Because researchers usually need a big pool of candidate DE genes from which they can choose the real DE genes based on biological justification, controlling FWER is too strict for this purpose. In contrast, the FDR approaches typically rejects more null hypotheses than the FWER approaches (Yekutieli and Benjamini (1999) and Benjamini and Yekutieli (2001)).

To sum up, the major problem in microarray data analysis is how to detect DE genes and control FDR. Numerous methods have been proposed on this subject. In Chapter 2, I will do a literature review of the related works.

1.3. Research Objectives

In this dissertation, the following research objectives will be addressed:

(1) To develop a t-mixture model approach to detect DE genes (Chapter 3).

(2) To develop a t-mixture model based FDR estimator (Chapter 4).

(3) To develop a two-step procedure to improve the current permutation based FDR estimator (Chapter 5).

(4) To develop a novel estimator of the proportion of EE genes (π_0; (1.3)), an important component in the FDR estimators. (Chapter 6).

CHAPTER 2

LITERATURE REVIEW

2.1. Detecting DE genes

Microarray was first used for gene expression profiling in Schena *et al.* (1995). The simplest method for detecting DE genes is the fold change method, which identifies a gene as DE if the expression level difference between two conditions is greater than some cut-off. The fold change method does not perform well because it ignores the different signal to noise ratio among different genes. Hence, more advanced microarray analysis techniques have been developed. They can be organized into two categories: parametric methods and nonparametric methods.

2.1.1. Parametric methods

One of the traditional parametric methods for detecting DE genes is the two sample t-test and its variations. Thomas *et al.* (2001) proposed to calculate the Z-score of each gene, which is the mean difference between two conditions divided by the pooled standard error of a gene, after correcting the sample heterogeneity using a regression approach. After that, the corresponding p-values are computed under asymptotic normality. Since the number of replicates is usually small for microarray data, the asymptotic normality can be strongly violated.

Other parametric methods have also been proposed. Newton *et al.* (2001) derived a hierarchical model for the gene expression levels. This model is based on the assumption that the

distribution of the mRNA intensity levels is Gamma. To identify the DE genes, the posterior odds of change is calculated. A gene will be considered as DE if the odds is too big or too small.

Kerr *et al*. (2000) proposed to use an ANOVA (analysis of variance) model which includes gene effect, array effect, their interaction effect. However, by using ANOVA, it implicitly assumes equal variance among genes, which is not appropriate. In contrast, Smyth (2004) constructed a linear model for the expression levels for every gene i. Suppose that Y_{ij}'s are defined as in Definition 1.1, then The proposed model is

$$Y_{ij} = a_i + b_i x_j + e_{ij},$$

where x_j=0 when $j \leq j_1$; x_j=1 when $j > j_1$; $var(e_{ij}) = \sigma_i^2$. The hypothesis now is to test whether b_i is significantly different from 0. The regular t statistic was used to test this hypothesis:

$$t_i = \frac{\hat{b}_i}{\tilde{s}_i \sqrt{v_i}},$$

where $\hat{b}_i$ is the least square estimate for b_i; v_i is some constant. $\tilde{s}_i^2$ is the estimator for σ_i^2. In regular linear models, the estimator for σ_i^2 is usually the mean square error s_i^2. For microarray data, the number of genes is so large that the information contained in other genes can be helpful to get a better estimate of σ_i^2. Hence, Smyth (2004) assumed a prior distribution on σ_i^2

$$\frac{1}{\sigma_i^2} \sim \frac{1}{d_0 s_0^2} \chi^2_{d_0},$$

where d_0 and s_0 are constants. Then $\tilde{s}_i^2$ is calculated as the posterior mean for σ_i^2. The implementation of this method can be found in Bioconductor package *limma* (Smyth (2005)).

Despite the simplicity, the above parametric methods all require strong model assumptions, which are often violated in practice. To overcome the model dependence problem, numerous nonparametric approaches of detecting significant DE genes have been proposed in the literature.

2.1.2. Nonparametric methods

The fundamental idea of the nonparametric methods is that, instead of obtaining the null distribution of the test statistic (denote by Z) from a known parametric distribution family, one constructs a null statistic (denote by z) which has the null distribution of the test statistic.

Dudoit *et al*. (2002) used the Welch t statistic

$$Z_i = \frac{Y_{i(1)} - Y_{i(2)}}{\sqrt{s^2_{i(1)}/j_1 + s^2_{i(2)}/j_2}} \tag{2.1}$$

as the test statistic. Where the Y_{ij}'s are defined same as in Definition 1.1; $Y_{i(1)}, Y_{i(2)}$ and $s^2_{i(1)}, s^2_{i(2)}$ are the sample means and sample variances of the Y_{ij}'s under two conditions, respectively. To get the null statistic z, Dudoit *et al*. (2002) permuted the J replicates and computed the corresponding test statistic z_i^b for the bth permutation. Repeat this procedure for B times, then B sets of null statistics $z_i^1, z_i^2, ..., z_i^B$ are obtained. With the null statistics, the p-value for gene i can be calculated as:

$$p_i = \frac{\sum_{b=1}^{B} I(|Z_i| < |z_i^b|)}{B}. \tag{2.2}$$

Troyanskaya *et al*. (2002) compared three methods: a nonparametric t-test method with test and null statistics same as in Dudoit *et al*. (2002), a Wilcoxon rank sum test method, and an

ideal discriminator method. It was shown in this paper that the Wilcoxon rank sum test method is the most conservative among these three.

Tusher *et al*. (2001) proposed a method called Significance Analysis of Microarray (SAM), which is one of the most popular methods in microarray analysis nowadays. In this paper, the test statistic is define similarly as in (2.1):

$$Z_i = \frac{Y_{i(1)} - Y_{i(2)}}{\sqrt{(1/j_1 + 1/j_2)s_i^2} + s_0} \tag{2.3}$$

s_i^2 is the pooled variance, and s_0 is a fudge factor used in SAM to minimize the coefficient of variation. The corresponding null statistics z_i^b is computed in a similar way as in Dudoit *et al*. (2002), but using within condition permutation. To detect DE genes, all genes are ranked by the magnitude of their Z_i so that $Z_{(1)}$ is the largest test statistic and $Z_{(i)}$ is the ith largest test statistic. For bth set of null statistics, the same procedure is applied so that $z_{(i)}^b$ is the ith largest null statistic in bth set. The expected relative difference is then defined as $z_{(i)}^E = \sum_{b=1}^{B} z_{(i)}^b / B$. After that, a scatter plot of $Z_{(i)}$ vs. $z_{(i)}^E$ is plotted. In the scatter plot, some points are displaced from the $Z_{(i)} = z_{(i)}^E$ line with a distance greater than pre-specified threshold Δ. The corresponding genes will be identified as DE genes.

Broberg (2003) used a similar test statistic Z_i as SAM. However, unlike in SAM, here s_0 was selected to minimize the number of false positive genes for a given significance level α.

Another well accepted method is the Empirical Bayesian (EB) method proposed in Efron *et al*. (2001). The test statistic Z_i and null statistic z_i in this paper are computed similarly as in Tusher *et al*. (2001). They then construct a mixture model for the test statistic:

$$f(z) = p_0 f_0(z) + p_1 f_1(z).$$

In the equation above, p_0=Prob(A gene is EE) is to be estimated; p_1=Prob(A gene is DE); $f_0(z)$ is the density for the test statistics of EE genes, which is also the null density; $f_1(z)$ is the density for the test statistics of DE genes. $f_0(z)$ and $f_1(z)$ cannot be estimated directly but their ratio can be estimated. Using Bayes rule, we can get the posterior probability that a gene is DE given test statistic Z:

$$p_1(z) = 1 - p_0 f_0(z)/f(z).$$

The ratio $f_0(z)/f(z)$ can be estimated by logistic regression and p_0 can be estimated by its upper bound $min_z f_0(z)/f(z)$. In this way, the posterior p-value Prob(gene is DE|Z) can be computed for every gene.

McLachlan *et al.* (2005) also proposed a method based on mixture model. They first transformed the test statistics into standardized z scores. Then they fitted a normal mixture model on the transformed z scores. Using the fitted density, the posterior p-value's can be computed as in Efron *et al.* (2001).

Pan *et al.* (2003) suggested using the test statistic Z_i as in (2.1) and null statistic same as in SAM:

$$z_i = \frac{dY_{i(1)}/j_1 - dY_{i(2)}/j_2}{\sqrt{s^2_{i(1)}/j_1 + s^2_{i(2)}/j_2}}, \tag{2.4}$$

where $dY_{i(1)} = \sum_{j=1}^{j_1/2} Y_{ij} - \sum_{j=j_1/2+1}^{j_1} Y_{ij}$, $dY_{i(2)} = \sum_{j=j_1+1}^{j_1+j_2/2} Y_{ij} - \sum_{j=j_1+j_2/2+1}^{j_1+j_2} Y_{ij}$, and $s^2_{i(1)}$, $s^2_{i(2)}$ are the sample variances of the $Y'_{ij}s$ under the two conditions.

Under the normality assumption, the numerator and the denominator of Z_i in in (2.1) are independent. However, this independence does not hold for z_i in (2.4). This shows that the

distribution of z_i is not the same as the null distribution of Z_i (Zhao and Pan (2003)). Therefore, z_i is not the null statistic of Z_i. Zhao and Pan (2003) and Pan (2003) proposed several modifications to fix this problem. They proposed to divide the replicates of each gene under the same experimental condition into two parts. The following are their latest version of the test statistic and its null statistic (Pan (2003)):

$$Z^1 = \frac{\frac{\overline{Y}_{11}+\overline{Y}_{12}}{2} - \frac{\overline{Y}_{21}+\overline{Y}_{22}}{2}}{\sqrt{\frac{s_{11}^2/j_{11}+s_{12}^2/j_{12}}{4} + \frac{s_{21}^2/j_{21}+s_{22}^2/j_{22}}{4}}}, \tag{2.5}$$

$$z^1 = \frac{\frac{\overline{Y}_{11}-\overline{Y}_{12}}{2} + \frac{\overline{Y}_{21}-\overline{Y}_{22}}{2}}{\sqrt{\frac{s_{11}^2/j_{11}+s_{12}^2/j_{12}}{4} + \frac{s_{21}^2/j_{21}+s_{22}^2/j_{22}}{4}}}, \tag{2.6}$$

where $j_{11} = j_{12} = j_1/2$ if j_1 is even, and $j_{11} = j_{12} - 1 = (j_1 - 1)/2$ if j_1 is odd. j_{21} and j_{22} are similarly defined. The statistics $(\overline{Y}_{11}, s_{11}^2)$,$(\overline{Y}_{12}, s_{12}^2)$, $(\overline{Y}_{21}, s_{21}^2)$,$(\overline{Y}_{22}, s_{22}^2)$ are the sample mean and variances of the four partitions of the replicates of each gene under the two experimental conditions. Those four partitions are $(Y_{ij}, j = 1, ..., j_{11})$ and $(Y_{ij}, j = j_{11} + 1, ..., j_1)$ from condition 1; $(Y_{ij}, j = j_1 + 1, ..., j_1 + j_{21})$ and $(Y_{ij}, j = j_1 + j_{21} + 1, ..., j_1 + j_2)$ from condition 2. For simplicity, the gene index i has been dropped in (2.5) and (2.6).

Further improvements on the construction of the test and null statistics have been developed in (Zhang , 2006). Realizing the need to pool the sample variances under the same experimental condition, Zhang (2006) proposed the following test statistic and null statistic:

$$Z1 = \frac{\frac{\overline{Y}_{11}+\overline{Y}_{12}}{2} - \frac{\overline{Y}_{21}+\overline{Y}_{22}}{2}}{\sqrt{\frac{\frac{1}{j_{11}}+\frac{1}{j_{12}}}{4} s_1^2 + \frac{\frac{1}{j_{21}}+\frac{1}{j_{22}}}{4} s_2^2}}, \tag{2.7}$$

$$z1 = \frac{\frac{\overline{Y}_{11}-\overline{Y}_{12}}{2} + \frac{\overline{Y}_{21}-\overline{Y}_{22}}{2}}{\sqrt{\frac{\frac{1}{j_{11}}+\frac{1}{j_{12}}}{4}s_1^2 + \frac{\frac{1}{j_{21}}+\frac{1}{j_{22}}}{4}s_2^2}}, \tag{2.8}$$

where $j_{jk}, \overline{Y}_{jk}$ (i, k= 1, 2) are defined the same as in Pan's statistics, and

$$s_1^2 = \frac{\sum_{j=1}^{j_{11}}(Y_{ij} - \overline{Y}_{11})^2 + \sum_{j=j_{11}+1}^{j_1}(Y_{ij} - \overline{Y}_{12})^2}{j_1 - 2 + I(j_1 = 2)},$$

$$s_2^2 = \frac{\sum_{j=j_1+1}^{j_1+j_{21}}(Y_{ij} - \overline{Y}_{21})^2 + \sum_{j=j_1+j_{21}+1}^{j_1+j_2}(Y_{ij} - \overline{Y}_{22})^2}{j_2 - 2 + I(j_2 = 2)}$$

are the two pooled sample variances from the replicates under each condition. It was demonstrated in Zhang (2006) that the test and null statistics (2.7) and (2.8) provide improvements over (2.5) and (2.6).

Suppose the density functions of z^1 and Z^1 from (2.5) and (2.6) are respectively f_0 and f. Pan (2003) used a normal mixture model method (MMM) to estimate f_0:

$$f_0(z; \psi_g) = \sum_{i=1}^{g} \pi_i \phi(z; \mu_i, \Sigma_i), \tag{2.9}$$

where $\phi(.; \mu_i, V_i)$ denotes the normal density function with mean μ_i and variance Σ_i, and $\pi_i's$ are mixing proportions, g is the number of components, which can be selected adaptively. ψ_g denotes all the unknown parameters $(\pi_i, \mu_i, \Sigma_i)|i = 1, ...g$. Similarly, f can also be fitted by a normal mixture model. After f and f_0 are fitted, for any given Z^1, Pan et al. (2003) used a likelihood ratio test statistic $LR(Z^1) = f_0(Z^1)/f(Z^1)$ to test for DE genes. When $LR(Z^1)$ is less than a certain value c, the gene will be identified as DE. The cut-off point c is determined

such that:

$$\frac{\alpha}{n} = \int_{LR(z)<c} f_0(z)dz, \tag{2.10}$$

where α is the genome-wide significance level, and α/n is the gene-specific significance level under Bonferroni adjustment to multiple comparison.

2.2. Estimating FDR

Like any hypothesis testing problem, a microarray data analysis method needs to control the Type I error rate. As mentioned before, the FDR is now a common choice of the control criterion. One type of FDR is called the "local false discovery rat", which is just the posterior probability of a gene being EE (Newton *et al.* (2001), Smyth (2004), Efron *et al.* (2001), McLachlan *et al.* (2005)). Efron *et al.* (2001) proved that the local FDR will converge to the regular FDR in (1.1) when the number of genes goes to infinity.

For the regular FDR, $E(P)$ in (1.3) is usually estimated by the number of significant genes. Hence, the number of false positive genes when all genes are EE (FP^* in (1.3)) and the proportion of EE genes (π_0 in (1.3)) are two key components left to be estimated.

2.2.1. Estimating the number of false positive genes when all genes are EE

The most commonly used method of estimating FP^* is the permutation method. Suppose the test statistic is Z_i, $i = 1, ..., n$; the null statistic is z_i^b for the bth set of permutations, $b = 1, ..., B$; and the rejection region is R, which means for gene i, $Z_i \in R \Rightarrow$ gene i is DE, then the

permutation method will estimate FP^* as

$$\widehat{FP^*} = \frac{\#(z_i^b : z_i^b \in R)}{B}. \tag{2.11}$$

Most of the methods estimate FP^* using this permutation method. In SAM (Tusher *et al*. (2001)), the number of false positive genes in each permutation was computed by counting the number of genes exceeding the cut-off distance Δ, and the final estimate of FP^* was the average number of false positives genes in all B permutations. Broberg (2003) used the same method to estimate FP^*. In Pan (2003), after the cut-off c in (2.10) is determined, the number of false positive gene is estimated as

$$\widehat{FP^*} = \frac{\#(z_i^b : LR(z_i^b) < c)}{B}, \tag{2.12}$$

where $LR(z_i^b)$ is defined in the same way as in (2.10). As we can see, (2.12) follows exactly the same idea in (2.11).

In Storey and Tibshirani (2003), a similar method as Pan (2003) is used to estimate FP^*. The only difference is that in Storey and Tibshirani (2003), instead of using the likelihood ratio, the authors compared the absolute value of z_i^b with some cut-off.

Although the permutation method has been widely accepted, A number of papers has discussed the correction of the overestimation problem of the permutation method. Pan (2003), Zhao and Pan (2003), Guo and Pan (2005) and Zhang (2006)) all proposed modified test and null statistics to address this problem. Xie *et al*. (2005) used another way to solve the overestimation problem. In their paper, they used one condition microarray data for illustration, which can easily be extended to the two conditions situation. Similarly as in Definition 1.1 except there is only one condition, suppose Y_{ij} is the expression level for gene i in array j, which has

mean μ_i and variance σ_i^2. The null hypothesis is $H_0 : \mu_i = 0$. Define the Bernoulli variable B_{ij}: $B_{ij} = 1$ with probability 0.5 and $B_{ij} = -1$ with probability 0.5. Then $W_{ij} = B_{ij}Y_{ij}$ is the gene expression levels after permutation. Xie *et al*. (2005) derived the mean and variance of W_{ij} as

$$\begin{aligned} E(W_{ij}) =& E(B_{ij})E(Y_{ij}) = 0 \\ var(W_{ij}) =& E(var(B_{ij}Y_{ij} \mid B_{ij})) + var(E(B_{ij}Y_{ij} \mid B_{ij})) \\ =& \sigma_i^2 + \mu_i^2 \end{aligned}$$

Hence, for all genes, the permuted gene expression level always has mean 0. However, for DE genes ($\mu_i \neq 0$), the variance of the permuted expression levels is bigger than the original variance. Subsequently, their paper showed that the over-estimation of FDR is caused by the fact that the distribution of null statistics generated from the permutation method is more dispersed than the true null distribution of the test statistics. To solve the problem, they proposed to exclude the predicted DE genes from the estimation of FDR.

2.2.2. Estimating the proportion of EE genes (π_0)

A number of methods have been proposed to estimate π_0 and most of them are based on the distribution of p-values under the null hypothesis. For gene i, the null hypothesis is that gene i is EE, and a p-value (p_i) is computed. Notice that the p-values of EE genes are uniformly distributed and denote the distribution of p-values of DE genes by $h_1(p)$. It is reasonable to model the overall p-values as a mixture distribution with two components (McLachlan and Peel, 2000):

$$h(p) = \pi_0 * 1 + (1 - \pi_0)h_1(p). \tag{2.13}$$

In Pounds and Morris (2003), the authors proposed a method called BUM using a beta-uniform mixture distribution to approximate $h(p)$. Then they estimated π_0 as $\hat{\pi}_0 = \hat{h}(1)$, which assumed $h_1(1) = 0$ and is an upper bound of the true π_0.

Langaas and Lindqvist (2005) adopted the same assumption but used nonparametric maximum likelihood method to estimate $\hat{h}(p)$.

SPLOSH (Pounds and Cheng (2004)) uses a local regression technique (LOESS; Cleveland and Devlin (1988)) to fit $h(p)$ and gives $\hat{\pi}_0 = min_p \hat{h}(p)$ as the estimator, which is still an upper bound.

Storey and Tibshirani (2003) proposed the QVALUE method. Given a tuning parameter λ, QVALUE estimates π_0 by

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

It can be proved that $\hat{\pi}_0(\lambda) \rightarrow \hat{h}(1)$ as $\lambda \rightarrow 1$ (Dalmasso *et al.* (2005)), so QVALUE also overestimates π_0 like BUM and SPLOSH.

All these estimators work well if the following assumption holds: few p-values of DE genes are close to 1. Otherwise, if this assumption is strongly violated which will happen when DE and EE genes are not well separated, all of them will tend to overestimate. There are other methods not requiring this assumption. Allison *et al.* (2002) proposed a parametric method to estimate π_0. Dalmasso *et al.* (2005) proposed the LBE method based on the moments of p-values, which also only gives an upper bound of π_0. More recently, Lai (2007) proposed a moment based method which requires no distribution assumption. Unfortunately, his method only works well when there are enough replicates (>8).

As we can see from above, the commonly used π_0 estimators BUM, SPLOSH, QVALUE and LBE are all actually upper bounds of π_0.

Most of the current π_0 estimators are based on p-values because a p-value is a unified measurement of significance. However, as a result of using p-values, we may lose some nice properties, such as the symmetry and unimodality of the original test statistics from which the p-values are computed. As we know, the commonly used test statistics are t-type statistics, which are generally symmetrically distributed and the use of symmetry can be helpful in estimation of π_0.

In an interesting paper, Bordes *et al*. (2006) proposed a nonparametric method to estimate the parameters in a two component mixture model with an unknown component, assuming the unknown distribution is symmetric. The authors also tried to apply this method to microarray data by fitting a similar model as (2.13) to test statistics. Since the t-type test statistics (without absolute value) for upregulated and downregulated DE genes obviously have different distributions, they cannot be modeled into one component. Hence, the authors constructed an F-type test statistic and assumed that it has a symmetric density. However, assuming an F-type test statistic to be symmetrically distributed is obviously not a reasonable assumption.

CHAPTER 3

THE T-MIXTURE MODEL APPROACH FOR DETECTING DIFFERENTIALLY EXPRESSED GENES IN MICROARRAYS

The finite mixture model approach has attracted much attention in analyzing microarray data due to its robustness to the excessive variability which is common in the microarray data. Pan *et al*. (2003) proposed to use the normal mixture model method (MMM) to estimate the distribution of a test statistic and its null distribution. However, considering the fact that the test statistic is often of t-type, our studies find that the rejection region from MMM is often significantly larger than the correct rejection region, resulting an inflated type I error. This motivates us to propose the t-mixture model (TMM) approach. In this chapter, we demonstrate that TMM provides significantly more accurate control of the probability of making type I errors (hence of the familywise error rate) than MMM. Finally, TMM is applied to the well-known leukemia data of Golub *et al*. (1999). The results are compared with those obtained from MMM.

3.1. Introduction

The use of microarray technology makes it possible to monitor the expression levels of thousands of genes simultaneously. A common goal of analyzing the genomewide expression data generated from this technology is to detect genes with differential expression under two conditions. Now, as the cost of microarray experiments keeps decreasing, replicated microarray

experiments are feasible. The replicated measurements of expression levels form the basis of the methods in this chapter.

In recent years, numerous nonparametric approaches for detecting significantly differentially expressed (DE) genes have been proposed in the literature (Efron *et al*. (2001); Tusher *et al*. (2001); Pan *et al*. (2003); Zhang (2006)), among others. In these nonparametric methods, the null distribution (the distribution of the test statistic for equivalently expressed (EE) genes) is estimated directly from the repeated measurements of gene expression levels under each condition.

In the mixture model method (MMM; Pan *et al*. (2003)), finite normal mixture models are used to estimate the distribution of the test statistic and the null distribution. However, noticing the fact that both the test and null statistics are usually heavy-tailed in practice, it is more natural to view them as the observations from a mixture of the t distributions. As pointed out in McLachlan and Peel (2000), the estimates of the component means and variances can be affected by observations that are atypical of the components in a finite normal mixture model. As a result, MMM may underfit the true underlying densities and produce critical values too small in absolute values. If the significance level approach is used, this will produce inflated type I error rates and lead to inflated familywise error rate (FWER).

To avoid the underfit problem of the MMM, some alternatives have been proposed in the literature (Allison *et al*. (2002); McLachlan *et al*. (2005)). The t-mixture model (TMM) approach has been proposed as a heavy-tailed alternative to the normal mixture model by McLachlan *et al*. (2002) for clustering the microarray-expressed data. However, the use of the TMM for the detection of DE genes was not discussed in their paper. In this chapter, we propose to use the TMM approach to estimate the distributions of the test statistic and its corresponding

null distribution. Following the lines of Pan *et al*. (2003), the null distribution is estimated from the permuted sets of null scores. We will show that the TMM can adapt to the atypical observations better than the MMM and provide more accurate critical values. In addition, our simulations show that no obvious improvement can be made by applying the TMM on more than one permuted set of null scores. Finally, we further illustrate the difference between the TMM and MMM by applying them to the leukemia data of Golub *et al*. (1999).

3.2. Methods

3.2.1. The test statistic and the null statistic

Suppose that Y_{ij} is the expression level of gene *i* in array j ($i = 1, 2, ..., n$; $j = 1, ..., j_1, j_1$+1,..., j_1+j_2), and the first j_1 and last j_2 arrays are obtained under the two different conditions. A general statistical model is

$$Y_{ij} = a_i + b_i x_j + \epsilon_{ij} \tag{3.1}$$

where $x_j = 1$ for $j \leq j_1$ and $x_j = 0$ for $j > j_1$. So, testing whether the mean expression levels under the two conditions is equivalent to testing the following hypothesis: $H_0 : b_i = 0$ against $H_1 : b_i \neq 0$.

The standard two sample *t*-statistic for testing this hypothesis is:

$$Z_i = \frac{\overline{Y}_{i(1)} - \overline{Y}_{i(2)}}{\sqrt{s^2_{i(1)}/j_1 + s^2_{i(2)}/j_2}}, \tag{3.2}$$

where $\overline{Y}_{i(1)}$,$\overline{Y}_{i(2)}$ and $s^2_{i(1)}$, $s^2_{i(2)}$ are the sample means and sample variances of the Y_{ij}'s under two conditions, respectively. Under the normality assumption of Y_{ij}, the null distribution of Z_i is approximately t-distributed.

However, when the normality assumption is violated, the use of the t distribution is not appropriate. A class of nonparametric statistical methods has been proposed to overcome this problem. The basic idea of the nonparametric methods is to estimate the null distribution of the test statistic Z by treating the values of the test statistic, when being applied to the permuted microarray data, as the true null scores one would expect from EE genes. However, recent research reveals that such practice is problematic. Zhao and Pan (2003) showed that one needs to modify the test statistic Z and construct its corresponding null statistic such that the null statistic, when being applied to the permuted microarray data, provides the correct null scores. Several methods for constructing the test statistic and the null statistic were proposed in Zhao and Pan (2003). However, it was pointed out in Pan (2003) that the methods of Zhao and Pan (2003) are quite restrictive. For example, it requires an even number of observations under each experimental condition. Improvements over Zhao and Pan (2003) were made in Pan (2003) in which he proposed the following test statistic and its corresponding null statistic:

$$Z^1 = \frac{\frac{\overline{Y}_{11}+\overline{Y}_{12}}{2} - \frac{\overline{Y}_{21}+\overline{Y}_{22}}{2}}{\sqrt{\frac{s^2_{11}/j_{11}+s^2_{12}/j_{12}}{4} + \frac{s^2_{21}/j_{21}+s^2_{22}/j_{22}}{4}}}, \tag{3.3}$$

$$z^1 = \frac{\frac{\overline{Y}_{11}-\overline{Y}_{12}}{2} + \frac{\overline{Y}_{21}-\overline{Y}_{22}}{2}}{\sqrt{\frac{s^2_{11}/j_{11}+s^2_{12}/j_{12}}{4} + \frac{s^2_{21}/j_{21}+s^2_{22}/j_{22}}{4}}}, \tag{3.4}$$

where $j_{11} = j_{12} = j_1/2$ if j_1 is even, and $j_{11} = j_{12} - 1 = (j_1 - 1)/2$ if j_1 is odd. j_{21} and j_{22} are similarly defined. The statistics $(\overline{Y}_{11}, s_{11}^2)$,$(\overline{Y}_{12}, s_{12}^2)$, $(\overline{Y}_{21}, s_{21}^2)$,$(\overline{Y}_{22}, s_{22}^2)$ are the sample mean and variances of the four partitions of the replicates of each gene under the two experimental conditions. Noticing the fact that the observations under the same condition are from the same population, Zhang (2006) provided an improved version of the above test statistic and null statistic:

$$Z1 = \frac{\frac{\overline{Y}_{11}+\overline{Y}_{12}}{2} - \frac{\overline{Y}_{21}+\overline{Y}_{22}}{2}}{\sqrt{\frac{\frac{1}{j_{11}}+\frac{1}{j_{12}}}{4} s_1^2 + \frac{\frac{1}{j_{21}}+\frac{1}{j_{22}}}{4} s_2^2}}, \tag{3.5}$$

$$z1 = \frac{\frac{\overline{Y}_{11}-\overline{Y}_{12}}{2} + \frac{\overline{Y}_{21}-\overline{Y}_{22}}{2}}{\sqrt{\frac{\frac{1}{j_{11}}+\frac{1}{j_{12}}}{4} s_1^2 + \frac{\frac{1}{j_{21}}+\frac{1}{j_{22}}}{4} s_2^2}}, \tag{3.6}$$

where $\overline{Y}_{jk}$ (i, k= 1, 2) are defined the same as in (3.3) and (3.4), and

$$s_1^2 = \frac{\sum_{j=1}^{j_{11}}(Y_{ij} - \overline{Y}_{11})^2 + \sum_{j=j_{11}+1}^{j_1}(Y_{ij} - \overline{Y}_{12})^2}{j_1 - 2 + I(j_1 = 2)} \tag{3.7}$$

$$s_2^2 = \frac{\sum_{j=j_1+1}^{j_1+j_{21}}(Y_{ij} - \overline{Y}_{21})^2 + \sum_{j=j_1+j_{21}+1}^{j_1+j_2}(Y_{ij} - \overline{Y}_{22})^2}{j_2 - 2 + I(j_2 = 2)} \tag{3.8}$$

are the two pooled sample variances from the replicates under each condition.

3.2.2. The t-mixture model

In the MMM, Pan et al. (2003) used a normal mixture model to estimate the density functions of Z^1 and z^1 defined by (3.3) and (3.4) and denoted them by f and f_0, respectively. As mentioned in the Section 3.1, it is more reasonable to view the test and null statistics as the observations

from a t-mixture model. In the TMM, it is assumed that the data are from several components with distinct t-distributions. That is, both f and f_0 are considered to be a mixture of the t distributions with probability density function:

$$h(z;\psi_g) = \sum_{i=1}^{g} \pi_i \varphi(z;\mu_i,\Sigma_i,\nu_i), \tag{3.9}$$

where $\varphi(z;\mu_i,\Sigma_i,\nu_i)$ denotes the t distribution density function with mean μ_i, variance Σ_i, and degrees of freedom ν_i. The coefficients π_i's are the mixing proportions and g is the number of components, which can be selected adaptively. ψ_g denotes all the unknown parameters $(\pi_i,\mu_i,\Sigma_i,\nu_i)|i=1,...g$ in (3.9).The TMM is fitted by maximum likelihood using an expectation conditional maximization (ECM) algorithm (Liu and Rubin (1995)). In the ECM algorithm, ψ_g is partitioned as (ψ_1^T,ψ_2^T), with $\psi_1^T = (\pi_i,\mu_i,\Sigma_i|i=1,...g)$ and $\psi_2^T = (\nu_i|i=1,...g)$. Given n p-dimension observations $y_j, j=1,...n$, on the $(k+1)th$ iteration of the ECM algorithm, the estimates of all the parameters are updated in two steps:

(1)

$$\pi_i^{(k+1)} = \sum_{j=1}^{n} \tau_{ij}^{(k)}/n,$$

$$\mu_i^{(k+1)} = \sum_{j=1}^{n} \tau_{ij}^{(k)} u_{ij}^{(k)} y_j / \sum_{j=1}^{n} \tau_{ij}^{(k)} u_{ij}^{(k)},$$

$$\Sigma_i^{(k+1)} = \frac{\sum_{j=1}^{n} \tau_{ij}^{(k)} u_{ij}^{(k)} (y_j - \mu_i^{(k+1)})(y_j - \mu_i^{(k+1)})^T}{\sum_{j=1}^{n} \tau_{ij}^{(k)}},$$

where

$$\tau_{ij}^{(k)} = \frac{\pi_i^{(k)} \varphi(y_j;\mu_i^{(k)},\Sigma_i^{(k)},\nu_i^{(k)})}{\sum_{i=1}^{g} \pi_i^{(k)} \varphi(y_j;\mu_i^{(k)},\Sigma_i^{(k)},\nu_i^{(k)})},$$

and

$$u_{ij}^{(k)} = \frac{\nu_i^{(k)} + p}{\nu_i^{(k)} + (y_j - u_i^{(k)})^T \Sigma_i^{(k)^{-1}} (y_j - u_i^{(k)})}.$$

(2) Find $\nu_i^{(k+1)}$ as a solution of the equation:

$$\begin{aligned} -digamma(\frac{\nu_i}{2}) + log(\frac{\nu_i}{2}) + 1 \\ + \frac{1}{\sum_{j=1}^n \tau_{ij}^{(k+1/2)}} \sum_{j=1}^n \tau_{ij}^{(k+1/2)} (log(u_{ij}^{(k+1/2)}) - u_{ij}^{(k+1/2)}) \\ + digamma(\frac{\nu_i(k) + p}{2}) - log(\frac{\nu_i(k) + p}{2}) = 0 \end{aligned}$$

where

$$\tau_{ij}^{(k+1/2)} = \frac{\pi_i^{(k+1)} \varphi(y_j; \mu_i^{(k+1)}, \Sigma_i^{(k+1)}, \nu_i^{(k)})}{\sum_{i=1}^g \pi_i(k+1) \varphi(y_j; \mu_i^{(k+1)}, \Sigma_i^{(k+1)}, \nu_i^{(k)})},$$

and

$$u_{ij}^{(k+1/2)} = \frac{\nu_i^{(k)} + p}{\nu_i^{(k)} + (y_j - u_i^{(k+1)})^T \Sigma_i^{(k+1)^{-1}} (y_j - u_i^{(k+1)})}.$$

At convergence, we obtain ψ_g^∞ as the maximum likelihood estimate. Since the ECM may give local maxima instead of global maxima, it is desirable to run this algorithm multiple times with different initial values and choose the estimates corresponding to the largest likelihood.

Another issue with the TMM is the determination of the number of components g. Here, we select the Bayesian Information Criterion (BIC) as the model selection criterion:

$$BIC = -2log(h(z; \hat{\psi}_g)) + t_g log(n),$$

where $h(z; \hat{\psi}_g)$ is defined in (3.9), and t_g is the number of independent parameters in the probability density function.

Due to the fact that the values of the test and null statistics in microarray analysis are usually heavy-tailed, we expect to see better performance of the TMM than the MMM when the given significance level is very small. We will verify this in the next section.

3.3. Results

3.3.1. Simulated data

Simulation set-ups To study the control of the type I error by MMM and TMM, we first consider the situation in which the null hypothesis holds. In this case, the expression levels of genes under the two experimental conditions are drawn from the same distribution. Two types of distributions are used: the standard normal and the t distribution with $df = 3$, representing the most commonly used distribution and a heavy-tailed distribution, respectively. The sample sizes are $j_1 = 4$ and $j_2 = 4$ for each gene, reflecting small sample sizes which are common in many microarray experiments. Here are the steps of the simulations.

(1) For each distribution, 10,000 genes are generated.

(2) We estimate f_0 with both the TMM and MMM. Only f_0 is considered in this chapter because the simulated data under the two conditions are from the same distribution.

(3) For each of the two estimates of f_0, a rejection region $z : |z| > z_0$ was established such that $P(|z| > z_0) = \alpha$, where α is the given significance level. In our study, α= 0.005, 0.001, 0.0005, 0.0001, 0.00005, 0.00001, 0.000005. Function uniroot( ) in R (R Development Core Team (2008)) is used to find z_0. In this way, we can get $z_0 = z_0^T$ with f_0 fitted by the TMM and $z_0 = z_0^M$ with f_0 fitted by the MMM. Finally, out of all gene i ($i = 1, ...10000$), we counted the proportion of genes with a corresponding

$|Z_i| > z_0^T$ and the proportion of genes with a corresponding $|Z_i| > z_0^M$, which are the Type I error rates for the TMM and MMM, respectively.

We repeated Steps 1 - 3 using both z^1 (Pan *et al*. (2003)) and $z1$ (Zhang (2006)) 100 times. Table 3.1 summarizes the average type I error rates for the TMM and MMM. We find that the MMM gives severely inflated type I error rates compared to the specified α when α is small and the TMM gives more accurate estimates. However, for the set-up with the t distributed data, we notice that when α is greater than 0.0001, the TMM is outperformed by the MMM, which motivates us to see how well the TMM and MMM fit the null statistics by checking the QQ plot between them. In Fig. 3.1, we can see that within a certain distance from 0, the TMM has a greater departure from the reference line comparing to the MMM. This is why the TMM gives higher false-positive rates and may even have larger variation for the false-positive rates than the MMM. However, this problem of the TMM is limited to the case when the data are from the t distribution and when the level of significance is relatively high. Hence, it is usually not a problem for the analysis of microarray data. For example, if the genome-wide level of significance is chosen as 0.01, the gene-specific level for a microarray data of 5,000 genes from the Bonferroni correction is 0.01/5000(=0.000002), which is much smaller than 0.0001, below which we found the TMM outperforms the MMM in our simulations. It can also be seen in Figure 3.1 that, when it comes to the tail, the TMM tends to stay closer to the reference line, and as the significance level decreases, its performance starts to improve and becomes significantly better than the MMM.

Another important factor which may affect the performance of the TMM is the stability of the estimates of the degree of freedoms. For the TMM, we found that the estimates of the degrees of freedom for the t-distributed data are not very stable, from 2.83 to 13.00. This is part

Table 3.1. Comparison of the MMM and TMM in Type I error rates at given gene specific levels of significance.

	Gene-specific	standard normal		t df=3	
Model	α	Pan	Zhang	Pan	Zhang
TMM	0.005	0.004988	0.004990	0.005962	0.006101
MMM		0.005105	0.005007	0.005049	0.005079
TMM	0.001	0.001049	0.001049	0.001589	0.00166
MMM		0.000990	0.000924	0.001135	0.001147
TMM	0.0005	0.000523	0.000523	0.000911	0.000946
MMM		0.000626	0.000569	0.000591	0.000589
TMM	0.0001	9.6e-05	9.6e-05	0.000246	0.000252
MMM		0.000264	0.00022	0.000221	0.000219
TMM	0.00005	4.2e-05	4.2e-05	0.00014	0.00014
MMM		0.000195	0.00016	0.000166	0.000165
TMM	0.00001	1e-05	1e-05	4.3e-05	4.1e-05
MMM		0.00012	9.4e-05	9.3e-05	9.2e-05
TMM	0.000005	6e-06	6e-06	2.7e-05	2.6e-05
MMM		9.8e-05	7.2e-05	7.6e-05	7.6e-05

of the reason why the TMM loses to the MMM when the levels of significance are relatively high.

Table 3.2 is obtained under similar set-up as that of Table 3.1 except now only the standard normal distribution is used. The purpose is to compare the performance of the TMM and MMM with respect to using only one permuted set of null scores and using all possible permuted sets of null scores (under the current setup, there are in total nine distinct permuted sets available). As we can see from Table 3.2, the actual type I error rates from MMM are severely inflated compared to the specified α values no matter how many permutations are there. In fact, there

are no significant differences between the results obtained from one set or nine sets of null scores for both the TMM and MMM. However, using nine permutations will cost much more computation time than just using one permutation.

We are also interested in the effect of the number of permutations in the presence of DE genes. For this purpose, we generated a total of 5,000 genes among which 200 were DE genes. The numbers of replicates under the two conditions are chosen as $j_1 = 4$ and $j_2 = 6$, respectively. For the first 100 DE genes, the data under condition 1 are generated from $N(0, 1)$ and the data under condition 2 are generated from $N(3, 1)$. For the remaining 100 DE genes, the data under condition 1 were generated from $N(0, 1)$ and the data under condition 2 were generated from $N(-1, 1)$. The data for EE genes are generated from $N(0, 1)$ under both conditions.

Same as Table 3.2, Table 3.3 shows the specified α levels and the observed Type I error rates of the TMM and MMM. In this case, the results from the TMM also show little difference between one set and thirty sets of null scores. However, there are significant changes in the results of the MMM as the number of permutations changes. For example, when the specified α is 0.00005, the observed type I error rate under one permutation is 0.00028 and is 0.00014 under thirty permutations. Although 0.00014 is still larger than 0.00005, it is much better than 0.00028; when the specified α is 0.00001, we have similar results. These results have two important implications. One is that when there exist DE genes, permutations of the null scores will help the MMM on fitting the heavy-tailed data. This is due to the over-dispersion of the permuted null scores (Xie *et al*. (2005)). The results also show that the number of permutations of null scores has little influence on the TMM, and the TMM performs consistently better than the MMM. Hence, the TMM is resistent to the over-dispersion problem of the null scores. Due

to the above observations, we suggest using the TMM with just one permutation. In this way, not only can we save a lot of computation time, but also we get better results.

Table 3.2. Comparison of the MMM and TMM in Type I error rates with respect to different number of permuted sets of null scores when all the genes are EE.

Gene-specific	One permutation		Nine permutations	
α	TMM	MMM	TMM	MMM
0.005	0.00490	0.00521	0.00509	0.00511
0.001	0.00103	0.00077	0.00102	0.00086
0.0005	0.00048	0.00046	0.00049	0.00045
0.0001	0.00009	0.00015	0.00010	0.00017
0.00005	0.00005	0.00010	0.00006	0.00013
0.00001	0.00001	0.00009	0.00001	0.00008

Table 3.3. Comparison of the MMM and TMM in Type I error rates with respect to different number of permuted sets of null scores and with the existence of DE genes.

Gene-specific	One permutation		Thirty permutations	
α	TMM	MMM	TMM	MMM
0.005	0.00479	0.00489	0.00485	0.00515
0.001	0.00093	0.00110	0.00094	0.00094
0.0005	0.00053	0.00066	0.00050	0.00051
0.0001	0.00010	0.00035	0.00010	0.00021
0.00005	0.00003	0.00028	0.00003	0.00014
0.00001	0.00000	0.00016	0.00001	0.00005

3.3.2. Leukemia data

The leukemia data of Golub *et al*. (1999) is one of the most studied gene expression data set. This data set includes 27 acute lymphoblastic leukemia (ALL) samples and 11 acute myeloid leukemia (AML) samples for 7129 genes. The goal is to find genes with differential expression between ALL and AML. Based on biological justification, Thomas *et al*. (2001) analyzed this

data set and identified 50 genes as the most expressed and related genes to the disease, including 25 most expressed genes for AML and 25 for ALL.

At each given genome-wide significance level α, we computed the cut-offs z_0^T for the TMM and z_0^M for the MMM. The Bonferroni method was used to adjust for multiplicity of the tests. Then, we calculated the test scores Z_i of the leukemia data and found the genes with $|Z_i| > z_0^T$ for the TMM and $|Z_i| > z_0^M$ for the MMM. These genes are the predicted DE or significant genes. Finally, we examined the predicted DE genes to see which method contains more genes from the Thomas *et al.* (2001) list of DE genes.

The results from the comparison are summarized in Table 3.4. When the genome-wide level α is 0.005, The TMM correctly identifies 35 out of 61 (53.38%) DE genes from the list while the MMM correctly identifies 37 out of 85 (38.32%) DE genes. Table 3.4 shows that the TMM consistently has a greater proportion of correctly identified genes than the MMM, which means that the TMM always has a smaller false positive rate, regardless of the levels of significance. Table 3.5 contains a list of the DE genes identified by both the TMM and MMM at $\alpha = 0.0005$. The p-values from Thomas *et al.* (2001) are given as the reference. As we expected, the MMM always gives smaller p-values than the TMM does. In other words, the MMM tends to provide smaller p-values due to its incapability to capture the true variability in the data and hence may contain more false-positive rates than TMM at the same level of significance.

3.4. Discussion

We have proposed to use the TMM for detecting the DE genes in a microarray experiment. Based on the simulation, the TMM can provide more accurate control of the probability of type I error than the MMM. Because the main focus of this chapter is to introduce the TMM

Table 3.4. Comparison of the results from the TMM and TMM at given levels of significance for the Leukaemia data.

Model	Genome-wide α	Total identified genes	Correctly identified genes	Correctly identified Proportion
TMM	0.05	130	42	0.3230
MMM		153	43	0.2810
TMM	0.01	77	37	0.4805
MMM		107	41	0.3832
TMM	0.005	61	35	0.5738
MMM		85	37	0.4353
TMM	0.001	37	27	0.7297
MMM		53	32	0.6038

Table 3.5. List of DE genes identified by both the TMM and MMM when genome-wide significance level is 0.0005.

Gene description	Probe	p-value		
		Thomas *et al.* (2001)	MMM	TMM
Macmarcks	HG1612–HT1612	<0.0001	2.57E-09	1.97E-07
Spectrin, alpha, nonerythrocytic 1 (alpha-fodrin)	J05243	<0.0001	2.67E-05	1.93E-04
IEF SSP 9502	L07758	<0.0001	6.52E-07	1.11E-05
Crystallin zeta (quinone reductase)	L13278	<0.0001	2.14E-05	1.63E-04
Inducible protein	L47738	<0.0001	3.52E-07	6.99E-06
Oncoprotein 18	M31303	<0.0001	1.36E-05	1.14E-04
Acyl-Coenzyme A dehydrogenase, C-4 to C-12 straight chain	M91432	<0.0001	1.04E-07	2.84E-06
CyclinD3	M92287	<0.0001	6.49E-07	1.11E-05
MB-1 (CD79b)	U05259	<0.0001	3.01E-06	3.55E-05
Cytoplasmic dynein light chain 1	U32944	<0.0001	3.17E-06	3.69E-05
Serine kinase SRPK2	U88666	<0.0001	2.94E-05	2.09E-04
Aldehyde reductase 1	X15414	<0.0001	1.23E-05	1.05E-04
Proteasome iota chain	X59417	<0.0001	7.13E-08	2.15E-06
p48	X74262	<0.0001	1.40E-10	2.63E-08
Adenosine triphosphatase, calcium	Z69881	<0.0001	9.46E-06	8.58E-05
Minichromosome maintenance deficient 3	D38073	<0.0001	5.54E-05	3.44E-04
Transcriptional activator hSNF2b	D26156	<0.0001	3.32E-06	3.82E-05
C-myb	U22376	<0.0001	4.01E-07	7.72E-06
Myosin light chain (alkali)	M31211	<0.0001	4.26E-08	1.47E-06
Transcription factor 3 (E2A)	M65214	<0.0001	1.20E-05	1.03E-04
Thymopoietin beta	U09087	<0.0001	1.43E-05	1.19E-04
Transcription factor 3 (E2A)	M31523	<0.0001	5.62E-06	5.73E-05
Fumarylacetoacetate	M55150	<0.0001	2.37E-12	1.98E-09

approach, we only discussed the control of false-positive rates by controlling the FWER. FWER can only provide control of the false-positive rates when no genes under consideration are DE. Hence, such control only works fine when there are none or very few genes which are actually

DE among all the genes in consideration. In the situations that a relatively substantial amount of genes are DE, more efficient control of the false positive rates can be achieved by controlling the false discovery rate (Benjamini and Hochberg (1995); Storey and Tibshirani (2003)).

Another point we want to stress is our proposal to only use one set of the permuted null scores when using the TMM. The current practice of the permutation-based methods often suggests using all possible permutations (or a subset of it if the total number of available permutations is too large). Such suggestions ignore the possible pitfalls which the correlated sets of permuted null scores could cause when using a method such as the EM algorithm (Dempster *et al*. (1977)) which requires i.i.d. observations. We believe that this proposal is important because not only can it significantly save computation time, which is the major concern of the finite mixture model approach, it can also avoid the problems caused by the use of the correlated permuted sets of null scores.

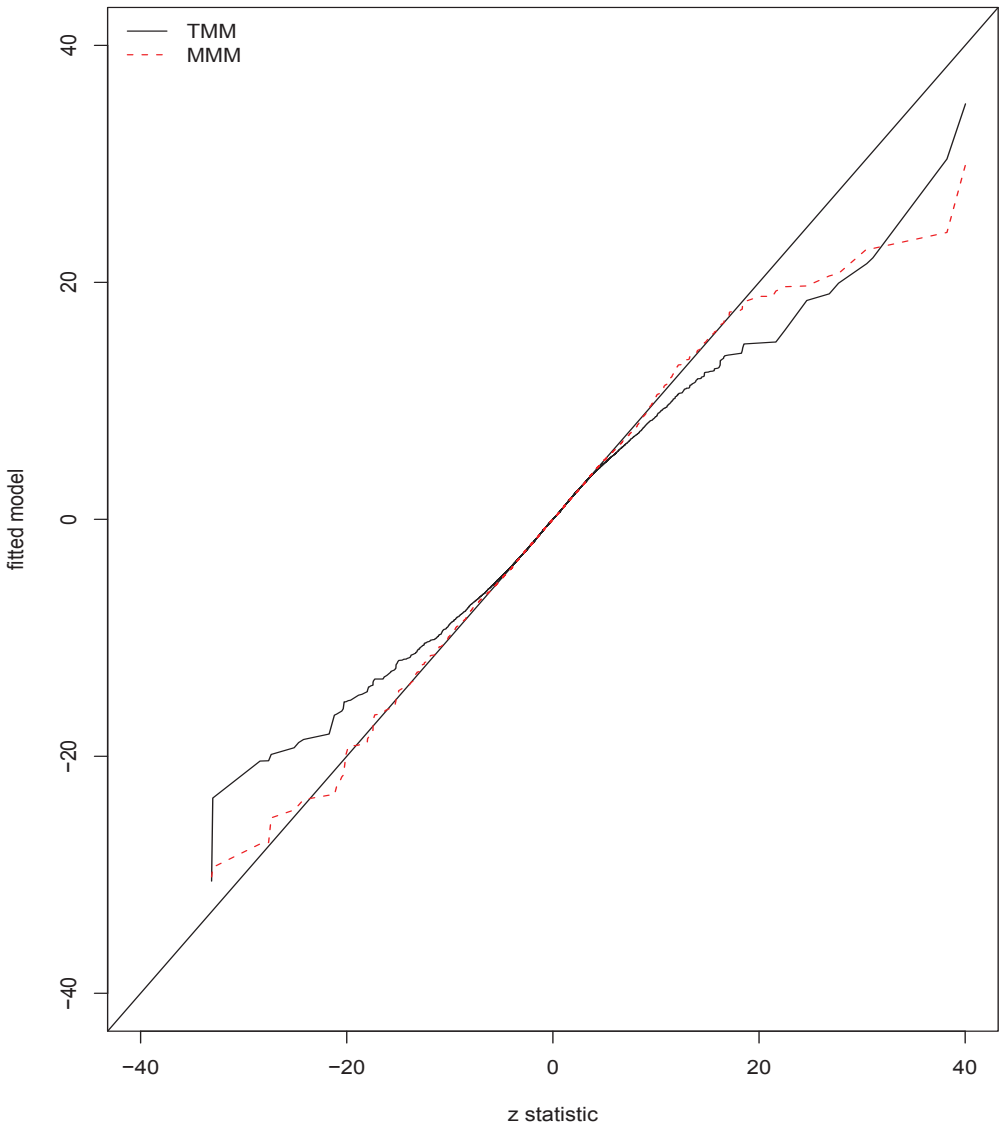

Figure 3.1. Plot of the comparison between TMM and MMM.

CHAPTER 4

A MIXTURE MODEL BASED APPROACH FOR ESTIMATING THE FDR IN REPLICATED MICROARRAY DATA

One of the most used methods for estimating the false discovery rate (FDR) is the permutation based method. The permutation based method has the well-known granularity problem due to the discrete nature of the permuted null scores. The granularity problem may produce very unstable FDR estimates. Such instability may cause scientists to over- or under-estimate the number of false positives among the genes declared as significant, and hence result in inaccurate interpretation of biological data. In this chapter, we propose a new model based method for estimating the FDR. The new method uses the t-mixture model which can model microarray data better than the currently used normal mixture model. We will show that our proposed method provides more accurate FDR estimates than the permutation based method and is free of the problems of the permutation based FDR estimators. Finally, the proposed method is evaluated using simulated and real microarray data.

4.1. Introduction

Genome-wide expression data generated from the microarray experiments are widely used to uncover the functional roles of different genes, and how these genes interact with each other. A key step to achieve this is to identify the differentially expressed (DE) genes under different experimental conditions. Such information can be used to identify disease biomarkers that may be important in the diagnoses of different types of diseases. Earlier statistical approaches for

detecting the DE genes focused mostly on parametric methods which are easily subject to model misspecification problems. Some of the well-known parametric methods for detecting DE genes include the two sample t-test (Long *et al.* (2001)), the analysis of variance approach (Kerr *et al.* (2000)), a regression approach (Thomas *et al.* (2001)), the parametric EB methods (Newton *et al.* (2001), Kendziorski *et al.* (2003)), and the linear model method (Smyth (2004)). Recently, the availability of replicated microarrays has made it possible to use the nonparametric methods to detect the DE genes. The nonparametric methods require much less stringent distributional assumptions, and thus can provide more robust results than the parametric methods. Some of the well-known nonparametric methods for analyzing microarrays include the Significance Analysis of Microarray (SAM) of Tusher *et al.* (2001), the nonparametric EB method (Efron *et al.* (2001)), the non-parametric t-test with adjusted p-value (Dudoit *et al.* (2002)), the Wilcoxon Rank Sum test (Troyanskaya *et al.* (2002)), samroc (Broberg (2003)) and the normal mixture model method (MMM) of Pan *et al.* (2003).

In this chapter, we will focus our attention on SAM, one of the most popular methods in microarray data analysis. SAM identifies DE genes by computing a modified t-statistic as the test score of a gene and finding the genes with test scores exceeding an adjustable threshold. The false discovery rate (FDR) was then estimated by a permutation based method. More specifically, the number of false positive (FP) genes among the significant genes is estimated as the median of the numbers of scores exceeding the cutoffs in each permuted set of null scores.

Since the permutation based approach estimates the FDR by counting the number of FP genes exceeding some cutoffs, we will call it the empirical method in this chapter. Due to its nature, there are two drawbacks with the empirical method: 1) the granularity problem – the FDR estimates based on the counted number of FP genes tend to be unstable when the actual

number of FP genes is small; 2) the zero FDR problem – the estimated FDR may be zero when the range of the permuted null scores is smaller than that of test scores and when the cutoffs are more extreme than the endpoints of permuted null scores. These two drawbacks are illustrated in the Figure 4.1, 4.2 and 4.3.

In this chapter, we will propose a t-mixture model based approach as an improvement of the empirical FDR estimation method of SAM. Our method aims to solve the two aforementioned drawbacks of the current empirical FDR estimation method: the granularity and the zero FDR problems. The performance of our method is assessed by applying them to simulated and real microarray data.

4.2. Methods

4.2.1. SAM

4.2.1.1. SAM algorithm. Let Y_{ij} be the expression levels of genes i under array j ($i = 1, ..., n$,$j = 1, ..., j_1, j_1 + 1, ..., j_1 + j_2 = J$), and the first j_1 and last j_2 arrays are obtained under two conditions. We need to test if gene *i* has differential expressions under the two conditions.

In SAM, the test statistic is defined as:

$$Z_i = \frac{Y_{i(1)} - Y_{i(2)}}{\sqrt{(1/j_1 + 1/j_2)s_i^2} + s_0},$$

where $Y_{i(1)}$ and $Y_{i(2)}$ are the sample means under two conditions; s_i^2 is the pooled sample variance; s_0 is the fudge factor. The null score z_i^b is then computed by applying the test statistic to the *b-th* set of permuted data.

In the SAM manual (Chu *et al*), the following algorithm is given to detect DE genes. First, all genes are ranked by the magnitude of their test scores Z_i so that $Z_{(1)}$ is the largest test score

and $Z_{(i)}$ is the *i-th* largest test score. For the *b-th* set of null scores, the same procedure is applied so that $z^b_{(i)}$ is the *i-th* largest null score in the *b-th* set of null scores. The expected relative difference is then defined as $z^E_{(i)} = \sum_{b=1}^{B} z^b_{(i)}/B$. After that, a scatter plot of $Z_{(i)}$ vs. $z^E_{(i)}$ is plotted. In the scatter plot, some points are displaced from the $Z_{(i)}$= $z^E_{(i)}$ line with a distance greater than Δ, a pre-specified threshold. Zhang (2007) pointed out that the estimated total number of significant (TS) genes and FP genes obtained using the SAM algorithm can be written as:

$$\widehat{TS} = \#\{i; Z_{(i)} > \delta_U \text{ or } Z_{(i)} < \delta_L\}, \tag{4.1}$$

and

$$\widehat{FP} = \sum_{b=1}^{B} \#\{i; z^b_{(i)} > \delta_U \text{ or } \mathrm{z}^b_{(i)} < \delta_L\}/B, \tag{4.2}$$

where δ_U and δ_L are the upper and lower cutoffs decided by the pre-specified threshold Δ. For simplicity, we only consider symmetric cutoffs ($|\delta_U| = |\delta_L|$) in this chapter though extensions to asymmetric cutoffs are straightforward. Under symmetric cutoffs, (4.1) and (4.2) can be written as:

$$\widehat{TS}(\delta) = \#\{i; |Z_{(i)}| > \delta\} \tag{4.3}$$

$$\widehat{FP}(\delta) = \sum_{b=1}^{B} \#\{i; |z_{(i)}^b| > \delta\}/B \tag{4.4}$$

4.2.1.2. Empirical FDR estimator of SAM. Given a gene-specific significance level $\alpha \in (0,\ 1]$ and assume that we have obtained the p-values for all the genes under consideration, the FDR proposed by Benjamini and Hochberg (1995) is defined as:

$$FDR = E[\frac{N(\alpha)}{TS(\alpha)}], \tag{4.5}$$

where $N(\alpha)$ is the number of genes among the EE genes whose *p*-values are less than or equal to α, and $TS(\alpha)$ is the number of genes among all the genes whose *p*-values are less than or equal to α (or it is the total number of significant genes). Instead of controlling gene-specific significance level α, SAM usually controls the total number of significant genes by setting a corresponding cutoff δ, hence (4.5) can be re-written as:

$$FDR = E[\frac{N(\delta)}{TS(\delta)}], \tag{4.6}$$

where $N(\delta)$ is the number of EE genes with $|Z_i|$ greater than δ, and $TS(\delta)$ is the total number of genes with $|Z_i|$ greater than δ.

It was shown in Storey and Tibshirani (2003) that the FDR can be approximated by

$$FDR \approx \frac{E[N(\delta)]}{E[TS(\delta)]}. \tag{4.7}$$

Since $N(\delta)$ is the number of false positives among the EE genes, denote the proportion of EE genes by π_0, (4.7) becomes

$$FDR \approx \frac{\pi_0 E[FP(\delta)]}{E[TS(\delta)]}, \tag{4.8}$$

where $FP(\delta)$ is the number of FP if all the genes are EE. $FP(\delta)$and $TS(\delta)$ can be estimated by $\widehat{FP}(\delta)$ and $\widehat{TS}(\delta)$ in (4.3) and (4.4), respectively. As a result, the empirical FDR estimator of SAM is

$$\widehat{FDR} = \frac{\hat{\pi}_0 \widehat{FP}(\delta)}{\widehat{TS}(\delta)}, \tag{4.9}$$

As mentioned before, this empirical FDR estimator of SAM has the granularity problem and the zero FDR problem. In the following sections, we solve these problems by proposing a model based FDR estimation method.

4.2.2. The t-mixture model (TMM) based FDR estimation approach

Let f be the probability density of the test score Z_i and f_0 be the density of null score z_i^b. In the TMM, it is assumed that the data are from several components with distinguished *t*-distributions. In other words, both f and f_0 are considered to be a mixture of the *t*-distributions with probability density function:

$$h(z;\psi_g) = \sum_{i=1}^{g} \pi_i \varphi(z;\mu_i,\Sigma_i,\nu_i), \tag{4.10}$$

where $\varphi(z;\mu_i,\Sigma_i,\nu_i)$ denotes the *t* distribution density function with mean μ_i, variance Σ_i, and degrees of freedom ν_i. The coefficients π_i are the mixing proportions and *g* is the number of components, which can be selected adaptively. ψ_g denotes all the unknown parameters (π_i,μ_i , Σ_i , ν_i) $|i = 1, ...g$ in (4.10). The mixture model is fitted by maximum likelihood using an expectation conditional maximization (ECM) algorithm (Liu and Rubin (1995)). The final model is selected based on the Bayesian Information Criterion (BIC). More details on how to fit the TMM to microarray data can be found in Jiao and Zhang (2008a). It was reported in their paper that not only does the TMM approach provide more accurate estimates of the densities, but also it enjoys computational efficiency since it was demonstrated in Jiao and Zhang (2008a) that one only needs to use one set of permuted null scores to fit the *t*-mixture model. More specifically, instead of using all z_i^b*'s (size=n*B)* to fit the *t*-mixture model, a random sample with size *n* can be drawn from $\bigcup_{b=1}^{B}\bigcup_{i=1}^{n} z_i^b$ and used as the null statistics.

Since the test statistic Z_i and the null statistic z_i (because only one set of null scores is used now, we will denote the null statistic as z_i instead of z_i^b) have the densities f and f_0, respectively,

it is easy to see from (4.8) that

$$
\begin{aligned}
FDR \approx & \pi_0 \frac{E[FP(\delta)/n]}{E[TS(\delta)/n]} \\
= & \pi_0 \frac{E \sum_{i=1}^{n} I(|z_i| \ \geq \delta)/n}{E \sum_{i=1}^{n} I(|Z_i| \ \geq \delta)/n} \\
= & \pi_0 \frac{P(|z| \geq \delta)}{P(|Z| \geq \delta)} \\
= & \pi_0 \frac{\int_{|z| \geq \delta} f_0(z) dz}{\int_{|z| \geq \delta} f(z) dz}, \qquad (4.11)
\end{aligned}
$$

where δ is chosen such that a given number of significant genes is detected. Equation (4.11) can be viewed as the model based formula of FDR.

Assume that we have available the estimators $\hat{f}$ and $\widehat{f_0}$ of f and f_0 from the TMM, respectively, then the corresponding model based FDR estimator for (4.11) is

$$
\widehat{\mathrm{FDR}}_1 = \widehat{\pi}_0 \frac{\int_{|z| \geq \delta} \widehat{f}_0(z) dz}{\int_{|z| \geq \delta} \widehat{f}(z) dz}, \qquad (4.12)
$$

The model based FDR estimator (4.12) has the following advantages compared to the empirical FDR estimator of SAM:

1) It does not have the granularity problem of the empirical FDR estimator (4.9);

2) It provides non-zero FDR estimate for any δ, while (4.9) only provides non-zero FDR when cutoffs are within the two endpoints of the range of the permuted null scores;

3) Unlike (4.9), the numerator and the denominator of (4.12) are not subject to the sampling variability.

4.3. Results

4.3.1. Simulated data

In the simulation, $j_1 = j_2 = 4$ replicates and n =5000 genes are generated while 200 of them are assumed to be differentially expressed. For the DE genes, the data under condition 1 are generated from $N(2,1)$ and the data under condition 2 are generated from $N(0,1)$. The EE genes are generated from $N(0,1)$ regardless of the conditions. For the generated data, we calculate the true FDR and estimated FDR for a grid of total number of significant genes ranging from 100 to 1 (in decreasing order). This procedure is repeated five times. Figure 4.1 shows comparisons of true FDR, empirical FDR estimator $\widehat{FDR}$ defined by (4.9), and the model based FDR estimator $\widehat{FDR}_1$ defined by (4.12).

As we can see, the instability of empirical $\widehat{FDR}$ increases significantly as it decreases to 0, which shows its granularity problem. Another fact worth noticing is that $\widehat{FDR}$ tends go to zero faster than the true FDR, which is the zero FDR problem. It can be seen that the true FDR strictly decreases as the total number of significant genes decreases. However, the empirical $\widehat{FDR}$ does not show this characteristic. In contrast, $\widehat{FDR}_1$ captures the decreasing trend very well and does not have the erratic jumps of $\widehat{FDR}$. To check how well these two FDR estimators approximate the true FDR, we calculate the mean squared error for both of them. MSE for $\widehat{FDR}$ is 0.00045 and MSE for $\widehat{FDR}_1$ is 0.00021, which shows that our method outperforms the empirical method.

Next, we compare the performances of the two methods when the two populations for the DE and EE genes are not so well separated. For this purpose, we conduct another simulation which tries to mimic the real data. The expression levels for the EE genes under

the two conditions are generated from $N(\mu_{i1}, \sigma_i^2)$ and $N(\mu_{i1}, \sigma_i^2)$ with $\mu_{i1} = \mu_{i2} \sim N(0, 2)$ $\sigma_i^2 \sim Gamma(4, 2)$. The expression levels for the DE genes are generated similarly as the EE genes, except that μ_{i1} and μ_{i2} are generated from $N(0, 2)$ separately. In this case, the grid of total number of significant genes ranges from 150 to 1 (in decreasing order). Comparison results are displayed in Figure 4.2.

It is seen from Figure 4.2 that $\widehat{FDR}$ is very unstable and approximates true FDR poorly, which makes the estimates highly inaccurate. On the other hand, $\widehat{FDR}_1$ has a much smoother curve than $\widehat{FDR}$ and seems to be able to capture the decreasing trend of the true FDR very well. In addition, the fact that MSE for $\widehat{FDR}$ is 0.025 and for $\widehat{FDR}_1$ is 0.015 shows that our method gives a significantly better fit to the true FDR.

4.3.2. Real data

The Leukemia data of Golub *et al.* (1999) is one of the most studied gene expression data sets. This data set includes 27 acute lymphoblastic leukemia (ALL) samples and 11 acute myeloid leukemia (AML) samples for 7129 genes. In Figure 4.3, we estimate the FDR for different number of significant genes using both our proposed model based FDR estimator and the empirical FDR estimator. As we expect, the model based FDR estimator gives a more stable estimate.

4.4. Discussion

In this chapter, we have proposed a *t*-mixture model based approach to improve the performance of SAM's empirical FDR estimator. We demonstrate that our method does not have the granularity and zero FDR problems as the empirical method. The results also show that

our estimator provides more stable and accurate estimates of the FDR. The advantage of our method is more evident in the case when DE genes are not well separated with EE genes and the variances of expression levels for every gene are different. This is due to the fact that the permutation FDR estimator is more easily affected by the sampling variability.

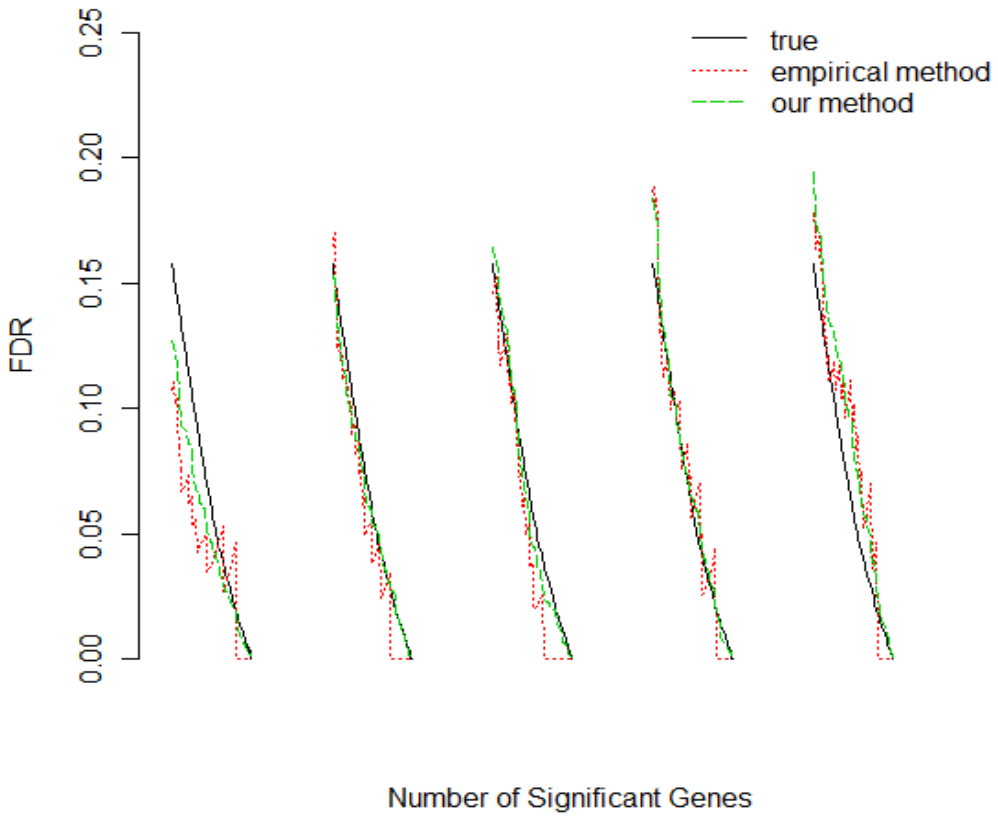

Figure 4.1. Comparison of the true FDR, the empirical FDR estimator $\widehat{FDR}$ and the model based FDR estimator $\widehat{FDR}_1$ for two sample microarray data. 5 replicates are listed. Total number of significant genes is decreasing from 100 to 1 (left to right) for each replicate.

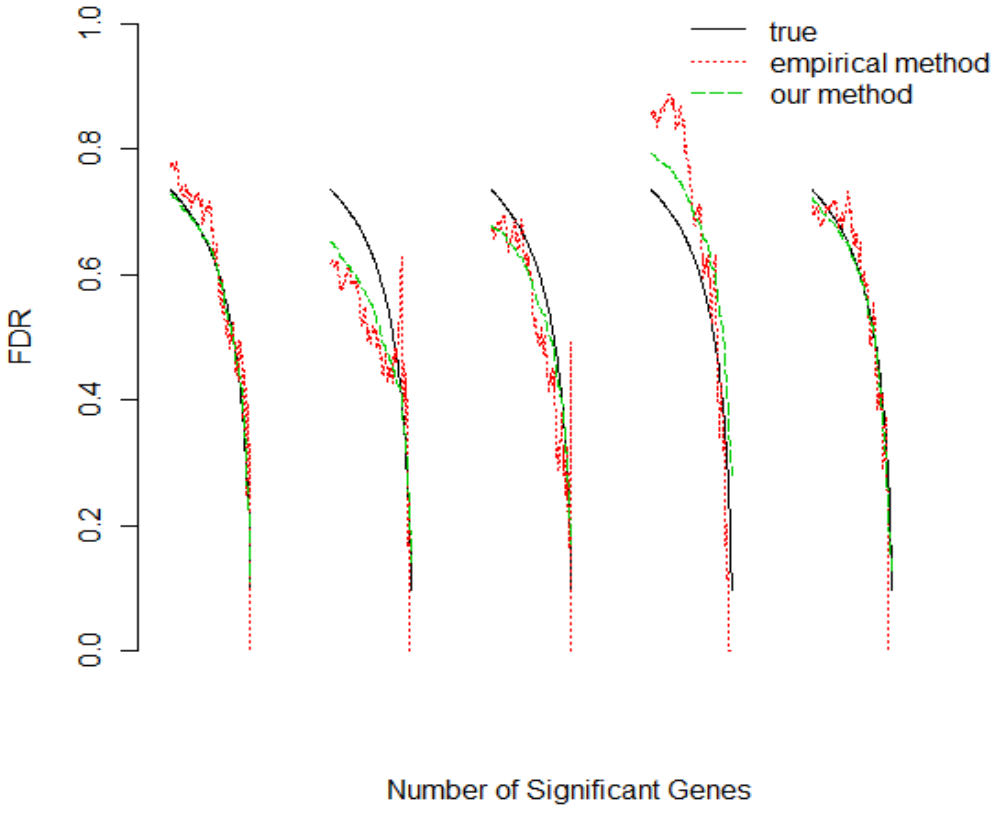

Figure 4.2. Comparison of the true FDR, the empirical FDR estimator $\widehat{FDR}$ and the model based FDR estimator $\widehat{FDR}_1$ for two sample microarray data. 5 replicates are listed. Total number of significant genes is decreasing from 150 to 1 (left to right) for each replicate.

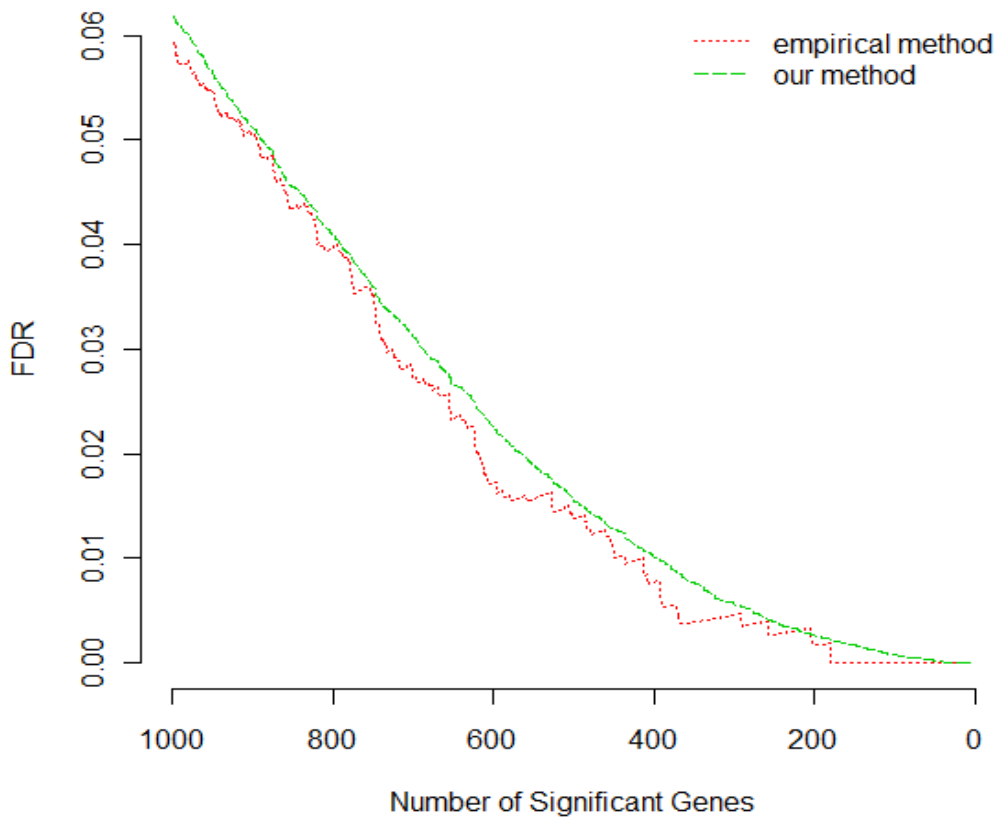

Figure 4.3. Comparison of the empirical FDR estimator $\widehat{FDR}$ and the model based FDR estimator $\widehat{FDR}_1$ for Leukemia microarray data.

CHAPTER 5

ON CORRECTING THE OVERESTIMATION OF THE PERMUTATION-BASED FDR ESTIMATOR

Recent attempts to account for multiple testing in the analysis of microarray data have focused on controlling the false discovery rate (FDR), which is defined as the expected percentage of the number of false positive genes among the claimed significant genes. As a consequence, the accuracy of the FDR estimators will be important for correctly controlling FDR. Xie *et al*. (2005) found that the overestimation of the FDR is caused by the discrepancy of the distribution of null statistics and the null distribution. More specifically, the distribution of null statistics from DE genes is more dispersed than the true null distribution. Since DE genes cause the problem, removing them seems to be an intuitive solution. Nevertheless, in practice true DE genes are unknown. Therefore, Xie *et al*. (2005) proposed to exclude the predicted DE genes from the estimation of FDR. However, we found that removing all the predicted DE genes is not a proper way to solve the problem. Other problems with their method include the biased estimation of FDR caused by over- or under- deletion of DE genes in the estimation of the FDR and by the implicit use of an unreasonable estimator of the true proportion of equivalently expressed (EE) genes. Due to the great importance of accurate FDR estimation in microarray data analysis, it is necessary to point out such problems and propose improved methods.

For this purpose, we propose a two-step procedure to estimate the FDR, in which the first step is to remove all the predicted DE genes and the second step is trying to re-include the

possible FP genes to construct the null statistics. Our results confirm that the standard permutation method overestimates the FDR. In addition, we show the method of Xie *et al*. (2005) always gives biased estimation of FDR: it overestimates when the number of claimed significant genes is small, and underestimates when the number of claimed significant genes is large. Most importantly, the results show that our two-step estimator gives more accurate FDR estimation.

5.1. Introduction

The use of microarray technology makes it possible to monitor the expression levels of thousands of genes simultaneously. A common goal of analyzing the genome-wide expression data generated from this technology is to detect DE genes. Now, as the cost of microarray experiments keeps decreasing, replicated microarray experiments are feasible.

Numerous methods (parametric and nonparametric) have been introduced to detect DE genes. Some of the most well known parametric approaches include the regression approach of Thomas *et al*. (2001), the empirical Bayes (EB) methods of Newton *et al*. (2001), Kendziorski *et al*. (2003), and the linear models and empirical Bayes methods of Smyth (2004). Among the nonparametric methods, some well known names include the EB method of Efron *et al*. (2001), the Significance Analysis of Microarray (SAM) of Tusher *et al*. (2001) and the mixture model method (MMM) of Pan *et al*. (2003).

The false discovery rate (FDR) introduced by Benjamini and Hochberg (1995) is now commonly used as the choice of the Type I error rate in microarray studies. It is defined as the expected percentage of false positive genes among the claimed significant genes. It was proved that in many cases controlling FDR is more appropriate compared to controlling family-wise error rate (FWER) since the FDR approaches typically reject more null hypotheses than

the FWER approaches (Yekutieli and Benjamini (1999) and Benjamini and Yekutieli (2001)). Several FDR controlling methods are implemented in the R *multtest* package (Pollard *et al*. (2004)).

However, the true FDR is unknown in practice. Hence, the estimated FDR will serve as the criterion to compare different methods when controlling the error rates. The comparison results are reasonable only if the estimated FDR approximates the true FDR well. The most common method of estimating the FDR is to use the permutation method. However, it has been reported in the literature that the permutation based FDR estimator tends to over- estimate the true FDR. A number of papers has discussed the correction of the over-estimation problem of the permutation method (Pan (2003), Zhao and Pan (2003), Guo and Pan (2005) and Zhang (2006)).

Xie *et al*. (2005) also noticed the overestimation problem of standard permutation method. Their paper showed that the over-estimation of the FDR is caused by the fact that the distribution of null statistics generated from the permutation method is more dispersed than the true null distribution of the test statistics. To solve the problem, they proposed to exclude the predicted DE genes from the estimation of the FDR. However, we find that their proposed method has serious under- or over-estimation problem depending on the number of genes declared significant. Another problem with the method of Xie *et al*. (2005) is that they implicitly used an estimator of the proportion of the EE genes (π_0) which can only provide good estimate of π_0 when the number of genes declared significant is equal or close to the true number of the DE genes in the microarray data and is otherwise biased.

5.2. Methods

5.2.1. The test statistics and the null statistics

As in Xie *et al*. (2005), only one-sample comparison will be considered in this chapter. Suppose that Y_{ij} is the expression level of gene *i* in array j ($i = 1, 2, ..., n$; $j = 1, ..., k$). The goal is to test the following hypothesis: $H_0 : E(Y_{ij}) = 0$ against $H_1 : E(Y_{ij}) \neq 0$. We use the same three test statistics as in Xie *et al*. (2005) for the purpose of comparison:

(1) The mean statistic: $M_i = \overline{Y}_i$,

(2) The *t*-statistic: $T_i = \frac{\overline{Y}_i}{V_i/\sqrt{k}}$,

(3) The SAM statistic: $S_i = \frac{\overline{Y}_i}{(V_i+V_0)/\sqrt{k}}$,

where $\overline{Y}_i = \sum_{j=1}^{k} Y_{ij}/k$, $V_i^2 = \sum_{j=1}^{k}(Y_{ij} - \overline{Y}_i)^2/(k-1)$, and V_0 is the fudge factor used to stabilize the variance.

In this chapter, we will focus on the permutation method for estimating the FDR. The key issue in the permutation method is the generation of the so-called null statistics (the values of the test statistic when the genes are EE). For convenience, we shall use Z_i as a general notation to denote the test statistic and use z_i to denote its corresponding null statistic. In the standard permutation method, one set of null statistics is calculated by applying the test statistic to one set of permuted data. The set of permuted data is obtained by randomly assign the "+" or "-" signs on each $Y_{i1}, ...Y_{ik}$ (SAM). Suppose the number of permutations is B, applying the test statistic to the *b*-th set of permutated data will create the *b*-th set of null statistics $z_i^{(b)}$, where $b = 1, ...B$, and $i = 1, ...n$.

5.2.2. Method for FDR estimation

Given the test statistics Z_i and a fixed cut-off value d, define $TS(d) = \#\{i : |Z_i| > d\}$ as the total number of significant genes; $FP(d) = \#\{i : |Z_i| > d, i \in EE\}$ as the number of false positive (FP) genes, where EE is the set of all equivalently expressed genes; π_0 as the proportion of EE genes; and $\hat{\pi}_0$ as its estimator. According to Storey and Tibshirani (2003), the false discovery rate can be approximated as

$$FDR(d) = E(\frac{FP(d)}{TS(d)}) \approx \frac{E(FP(d))}{E(TS(d))}. \tag{5.1}$$

A practical version of the FDR is the false discovery proportions (FDP) defined by

$$FDP(d) = \frac{FP(d)}{TS(d)}. \tag{5.2}$$

To estimate the FDR, the standard method is to use the permutated null statistics. Define

$$\widehat{FP}(d) = \sum_{b=1}^{B} \#\{i : |z_i^{(b)}| > d\}/B. \tag{5.3}$$

Notice that $\widehat{FP}(d)$ is actually an estimate of $FP(d)/\pi_0$. Storey and Tibshirani (2003) suggested to estimate the FDR by

$$\widehat{FDR}(d) = \frac{\hat{\pi}_0 \widehat{FP}(d)}{TS(d)}. \tag{5.4}$$

However, as shown in Xie *et al*. (2005), although the null statistics of the EE genes have the true null distribution of test statistics, the null statistics of DE genes are more dispersed than those of EE genes. As a result, the empirical distribution of the null statistics from all genes is not a good approximation to the true null distribution. To overcome this problem, Xie

et al. (2005) proposed a new FDR estimator. Their idea is as follows: Since the over-estimation problem of standard permutation method is caused by the DE genes, using only EE genes to construct the null distribution will avoid this problem. Nevertheless, in practice which genes are EE genes is unknown. Therefore, they proposed to use the predicted EE genes to estimate the FDR. Their FDR estimation procedure works as follows: Suppose Z_i is the test statistic and S_i is the SAM statistic, for any given $d > 0$, any gene i with $|S_i| > d$ is said to be significant. $TS(d)$ is defined the same as before. Define a set of non-significant genes $D(d) = \{i : |S_i| \leq d'\}$, where S_i is the SAM statistic and d' is chosen so that the number of genes not in set $D(d)$ is the same as $TS(d)$. In other words, $D(d) = \Omega - TS(d)$, where Ω is the set of all genes. $\widehat{FP}(d)$ is then estimated by constructing B sets of null statistics as before. The only difference is that only genes in $D(d)$ are going to be used this time. Let

$$\widehat{FP}(d)_0 = \sum_{b=1}^{B} \#\{i \in D(d) : |z_i^{(b)}| > d\}/B. \tag{5.5}$$

Then, the FDR is estimated by

$$\widehat{FDR}(d)_0 = \frac{\widehat{FP}(d)_0}{TS(d)}. \tag{5.6}$$

Note that $\widehat{FP}(d)_0$ in (5.6) is the average number of significant genes found from the genes in $D(d)$. We can re-write (5.6) in the form of (5.4) as

$$\widehat{FDR}(d)_0 = \hat{\pi}_0 \widehat{FP}(d)/TS(d), \tag{5.7}$$

where

$$\widehat{FP}(d) = \frac{n}{n - TS(d)} \widehat{FP}(d)_0 \tag{5.8}$$

can be viewed as the average number of significant genes if all n genes are EE and $\hat{\pi}_0 = 1 - TS(d)/n$ is the estimated proportion of EE genes in the microarray data.

In Xie *et al*. (2005), the above method was proved to be able to correct the FDR overestimation problem of the permutation method effectively. However, our study has found that (5.6) has some problems:

(1) It can be seen from (5.7) and (5.8) that Xie *et al*. (2005) implicitly uses $\hat{\pi}_0$=$1-TS(d)/n$ as an estimate of π_0. Noticing that $TS(d)$ is the number of claimed significant genes, such $\hat{\pi}_0$ can range from 0 to 1 for $TS(d)$ from n to 0. As a consequence, one will always under- or over- estimate π_0 unless $TS(d)$=the true number of DE genes.

(2) The over- or under- estimation of FDR due to under- or over- deletion of genes, which will be discussed in section 5.2.3.

(3) In Xie *et al*. (2005), the SAM statistic was used to define the set $D(d)$, which is used in (5.5) to estimate the number of FP even if the test statistic is the mean or *t*-statistics. This is unreasonable. If one has chosen the mean or t statistic as the test statistic, why would he/she use a different statistic to estimate the number of FP? The only explanation is that the mean statistic and the t-statistic do not provide results as good as the SAM statistic does. Note that the mean statistic and the t-statistic can be viewed as two extreme cases of the SAM statistic with the fudge factor equal ∞ and 0, respectively. It is well known that the performance of the testing procedure based on the mean statistic and the t-statistic is generally inferior to that based on the SAM statistic.

5.2.3. Our proposed method for FDR estimation

Considering the unreasonable estimates $\hat{\pi}_0$ of Xie *et al*. (2005) may provide, we suggest estimating π_0 by the method introduced in Storey and Tibshirani (2003), which is implemented in SAM. In their paper, they calculated p-values for each gene. Denote the p-values by $p_1, p_2, ...p_n$. Then, π_0 is estimated by $\hat{\pi}_0^{sam} = \#\{p_i > \lambda\}/(n(1-\lambda))$, where λ is a tuning parameter. As we can see, $\hat{\pi}_0^{sam}$ is a constant no matter how $TS(d)$ changes. In addition, unlike in Xie *et al*. (2005), we use the same test statistic for both identifying the DE genes and defining the set $D(d)$. In other words, $D(d) = \{i : |Z_i| \leq d\}$. With $\hat{\pi}_0^{sam}$ and this new $D(d)$, we propose the following FDR estimator

$$\widehat{FDR}(d)_1 = \hat{\pi}_0^{sam}\widehat{FP}(d)/TS(d), \tag{5.9}$$

where $\widehat{FP}(d) = \frac{n}{n-TS(d)}\widehat{FP}(d)_0$.

The estimator $\widehat{FDR}(d)_1$ corrects Xie *et al*'s method by using a more reasonable estimator of π_0. However, another question comes to light: Is removing all the predicted DE genes a proper way of estimating the FDR? As we know, what we really want is to remove all the DE genes and use all the EE genes to construct the null statistics. However, in those predicted DE genes, there are some genes which are actually EE genes, but are falsely identified as positive (FP genes). It is obvious that the FP genes are the EE genes with the greatest test statistics in absolute values. Therefore, excluding such genes will cause underestimation of the tail of the null distribution. In Section 5.3.2, we will show that removing all the predicted DE genes gives significantly different FP estimates from those obtained by removing the true DE genes (which is not feasible in practice but good for comparison).

Since removing all predicted DE genes will cause underestimation of the FDR, an intuitive solution would be to add the FP genes back into the pool of the genes for the estimation of the FDR. For this purpose, we propose the following two-step procedure to estimate the FDR, in which the first step is to remove all the predicted DE genes and the second step is trying to re-include the possible FP genes to construct the null statistics:

(1) Suppose Z_i is the test statistic, for any given $d > 0$, any gene i with $|Z_i| > d$ is said to be significant. Let $TS(d) = \#\{i : |Z_i| > d\}$, $D(d) = \{i : |Z_i| \leq d\}$, $\widehat{FP}(d)_0 = \sum_{b=1}^{B} \#\{i \in D(d) : |z_i^{(b)}| > d\}/B$, and $\widehat{FDR}(d)_1 = \frac{n}{n-TS(d)} \hat{\pi}_0^{sam} \widehat{FP}(d)_0 / TS(d)$.

(2) Using $\widehat{FDR}(d)_1$ from Step 1, let $D(d') = \{i : |Z_i| \leq d'\}$, d' is chosen such that the number of genes not in $D(d')$ is $TS(d') = TS(d)(1 - \widehat{FDR}(d)_1)$. Then following the same procedure as step 1, we get $\widehat{FP}(d')_0 = \sum_{b=1}^{B} \#\{i \in D(d') : |z_i^{(b)}| > d'\}/B$, and

$$\widehat{FDR}(d)_2 = \hat{\pi}_0^{sam} \widehat{FP}(d)/TS(d), \tag{5.10}$$

where $\widehat{FP}(d) = \frac{n}{n-TS(d')} \widehat{FP}(d')_0$.

The idea behind our proposed method is as follows: When the number of predicted DE genes is greater than the true number of significant genes, there will be a substantial number of FP genes in them. Since removing all predicted DE genes will cause biased estimation of the FDR, we only remove the genes which we consider are most likely to be true DE genes.

5.3. Results

5.3.1. Problems caused by using Xie *et al*'s estimate of π_0

In Xie *et al*. (2005), π_0 is estimated by $\hat{\pi}_0 = 1 - TS(d)/n$. As stated before, we would expect to see over- or under- estimation of FDR by this method because of the over- or under- estimation of π_0 by $\hat{\pi}_0$.

To show this, 5 $(= k)$ replicates of $4000(= n)$ genes are generated, among which 400 are DE genes and the others are EE genes. The expression levels Y_{ij} for EE genes are generated from $N(0, 4)$ and Y_{ij} for DE gene are generated from $N(\mu_i, 4)$, while $\mu_i \sim N(0, 16)$. The SAM, mean, and *t*-statistics are used as the test statistics. Our purpose is to compare the FDR estimator of Xie *et al*. (2005) ($\widehat{FDR}(d)_0$) from (7) and one of our proposed estimator ($\widehat{FDR}(d)_1$) from (5.9). The values of the standard FDR estimator from (4) and the true FDR values are also plotted as references.

5.3.1.1. Overestimation of FDR when $TS(d)$ is smaller than the true number of DE genes.

.

In this scenario, $TS(d)$ is set to vary between 100 and 200, which is much less than the true number of DE genes $(= 400)$. In Figure 5.1, as we expected, $\widehat{FDR}(d)_0$ always overestimates the true FDR while $\widehat{FDR}(d)_1$ provides less biased estimates. In some cases, $\widehat{FDR}(d)_1$ still gives overestimation. This overestimation is caused by the fact that the $\hat{\pi}_0^{sam}$ also always overestimates the true π_0, but to a much lesser degree.

5.3.1.2. Underestimation of FDR when $TS(d)$ is greater than the true number of DE genes.

.

The same simulation set-up is used as above except now $TS(d)$ is set to be vary between 500 to 600, which is greater than the true number of DE genes $(= 400)$.

As shown in Figure 5.2, for the *t* and SAM statistics, Xie *et al*'s method underestimates the true FDR while our proposed method gives more accurate estimates. However, for the mean statistic, our method does not give any improvement over Xie *et al*'s method. The reason is that the SAM statistic was used to predict DE genes in Xie *et al*. (2005) while our method $\widehat{FDR}(d)_1$ uses the same mean statistic in both predicting the DE genes and estimating the FDR. The better performance of Xie *et al*'s method in this case is due to the use of the SAM statistic in predicting DE genes, rather than the method itself. As it can be seen from the top plot of Figure 5.2, our estimator $\widehat{FDR}(d)_1$ performs much better than Xie *et al*'s method when the SAM statistic is used.

5.3.2. Underestimation caused by removing the predicted DE genes

In this section, we show that removing all predicted DE genes will lead to an underestimation of the true false positive number. We generate $n = 4000$ genes with $k = 5$, while 150 of them are DE genes. The expression levels for EE and DE genes are generated in the same way as in Section 5.3.1. The number of claimed significant genes is set to be 150, which is the number of true DE genes. Table 5.1 lists the true FP number, the estimated FP number with 150 predicted DE genes removed ($\widehat{FP}_p$), and the estimated FP number with 150 true DE genes removed ($\widehat{FP}_t$). The results reported are the averages from 50 replicates.

From Table 5.1, we can see $\widehat{FP}_p$ is always less than $\widehat{FP}_t$. This shows removing predicted DE genes gives a smaller estimate of FP number than that of removing the true DE genes.

Table 5.1. Comparison of estimated false positive numbers and the true false positive numbers using the SAM, mean and *t*-statistics. $\widehat{FP}_p$ is the estimated FP number with 150 predicted DE genes removed; $\widehat{FP}_t$ is the estimated FP number with 150 true DE genes removed.

Statistic	True FP	$\widehat{FP}_p$	$\widehat{FP}_t$
SAM	64.38	61.62	65.30
mean	58.96	53.96	60.81
t	79.78	77.21	81.19

5.3.3. Performance of our methods

To evaluate the performance of our methods, the same simulation set-ups are used as those in Section 5.3.2. We want to see whether our proposed estimator $\widehat{FDR}(d)_2$ from (5.10) can overcome the problems or at least has some advantages over other estimators.

We compare four different FDR estimation methods: the standard estimator $\widehat{FDR}(d)$ from (5.4), Xie *et al*. (2005)'s estimator $\widehat{FDR}(d)_0$ from (5.7), and two estimators we proposed: $\widehat{FDR}(d)_1$ from (5.9), $\widehat{FDR}(d)_2$ from (5.10).

Figure 5.3 shows that the estimator of Xie *et al*. (2005) always significantly underestimates the true FDR's. The estimator $\widehat{FDR}(d)_1$ also underestimates FDR due to over-deletion, but is much better than Xie *et al*'s estimator for the SAM statistic. For the mean and *t*-statistics, Xie *et al*'s estimator outperforms $\widehat{FDR}(d)_1$ sometimes due to the same reason discussed previously $-$ the use of the SAM statistic in obtaining the predicted DE genes. In contrast, $\widehat{FDR}(d)_2$ does not have this problem and has the best performance in most of the scenarios. However, for the SAM statistic and the *t*-statistic, $\widehat{FDR}(d)_2$ slightly overestimates the true FDR. This over-estimation is not caused by the the underlying algorithm of estimator $\widehat{FDR}(d)_2$, but by the overestimation of π_0 caused by $\hat{\pi}_0^{sam}$. To see this, we replaced $\hat{\pi}_0^{sam}$ in (5.9) ($\widehat{FDR}(d)_1$) and in (5.10) ($\widehat{FDR}(d)_2$) with the true $\pi_0 = 3850/4000$. Figure 5.4 shows the comparison between

the true FDR and the estimated FDR from (5.9) and (5.10) with the true value of π_0. We can see that $\widehat{FDR}(d)_2$ now gives smaller estimates of FDR for all three test statistics compared to Figure 5.3. Another fact worth noticing in Figure 5.3 and 5.4 is that when the number of claimed significant genes is small, $\widehat{FDR}(d)_2$ does not show much advantage. The reason is that, in such a case, most of the significant genes are true DE genes and the number of FP genes is much smaller than the number of true DE genes. Hence, removing the FP genes is not going to have significant impact on the estimation of the FDR.

5.3.4. Comparisons under other simulation set-ups

We also want to see how the ratio of induced (I) and repressed (R) genes influences the performance of the FDR estimators. Here, $k = 5$, $n = 4000$ and there are 150 DE genes. The expression level Y_{ij} for EE genes are generated from $N(0, 4)$. For DE genes, n' of them are generated from $N(4, 4)$, and the rest of them are generated from $N(-4, 4)$; where $n' = 150, 100, 50, 0$. We set the number of claimed significant genes as 300. The results reported in Table 5.2 are the averages from 50 replications. The results confirm that our methods are stable to the change of ratios of the induced and repressed genes.

We have also conducted another simulation which tries to mimic the real data. A similar simulation set-up as above is used except the expression level Y_{ij} for EE genes are generated from $N(0, \sigma_i^2)$ while $\sigma_i^2 \sim Gamma(4, 2)$ and Y_{ij} for DE gene are generated from $N(\mu_i, \sigma_i^2)$ while $\mu_i \sim N(0, 16)$, $\sigma_i^2 \sim Gamma(4, 2)$.

Table 5.2. Comparison of the performance of FDR estimator when the ratio of induced and repressed genes changes.

I/R		FDR_{true}	$\widehat{FDR}(d)$	$\widehat{FDR}(d)_0$	$\widehat{FDR}(d)_1$	$\widehat{FDR}(d)_2$
150/0	SAM	0.507	0.572	0.461	0.486	0.521
	mean	0.504	0.672	0.423	0.446	0.504
	t	0.558	0.564	0.513	0.539	0.560
100/50	SAM	0.508	0.566	0.463	0.489	0.520
	mean	0.504	0.665	0.416	0.439	0.498
	t	0.557	0.569	0.512	0.538	0.562
50/100	SAM	0.509	0.570	0.460	0.485	0.520
	mean	0.504	0.670	0.424	0.445	0.499
	t	0.557	0.565	0.512	0.537	0.558
0/150	SAM	0.507	0.566	0.465	0.491	0.522
	mean	0.504	0.661	0.427	0.449	0.504
	t	0.556	0.562	0.514	0.544	0.562

From Figure 5.5, we can see that the results are similar as before for the SAM and t statistics: the standard method always overestimates and method of Xie *et al*. (2005) always underestimates. $\widehat{FDR}(d)_1$ performs better than Xie *et al*. (2005)'s method and $\widehat{FDR}(d)_2$ always performs the best.

5.3.5. Biological Data

In Zhong *et al*. (2004), duplications and deletions in an evolved strain (DD2459) were identified by a whole-genome *E. coli* MG1655 spotted DNA microarray experiment with three replicates. 38 genes have been confirmed to be true duplicated/deleted genes by rtPCR. To compare our proposed estimator $\widehat{FDR}(d)_2$ with Xie *et al*. (2005)'s estimator $\widehat{FDR}(d)_0$, we used this dataset to construct a table summarizing the upper bound of true FDR (the proportion of detected DE genes which are not in the confirmed 38 DE genes), FDR estimates given by $\widehat{FDR}(d)_2$ and

Table 5.3. Comparison of the performance of $\widehat{FDR}(d)_2$ and $\widehat{FDR}(d)_0$ using microarray data from Zhong *et al*. (2004).

Statistic	TS(d)	Upper bound	$\widehat{FDR}(d)_0$	$\widehat{FDR}(d)_2$
SAM	35	0.457	0.347	0.506
	40	0.500	0.304	0.443
	45	0.533	0.272	0.404
	50	0.560	0.267	0.386
mean	35	0.371	0.230	0.356
	40	0.375	0.158	0.264
	45	0.422	0.171	0.242
	50	0.480	0.177	0.231
t	35	1.000	0.871	1.000
	40	1.000	0.870	1.000
	45	1.000	0.817	1.000
	50	1.000	0.810	0.997

$\widehat{FDR}(d)_0$ for different number of total significant genes ($TS(d)$). Because the confirmed 38 true DE genes are mostly genes with largest mean in absolute value, we can see from Table 5.3 that the mean statistic gives the smallest FDR upper bound while the *t*-statistic does not detect any one of the 38 true DE genes. Table 5.3 also shows that $\widehat{FDR}(d)_2$ always gives more accurate FDR estimates than $\widehat{FDR}(d)_0$.

5.4. Discussion

In this chapter, we have showed that the bias-corrected FDR estimator proposed in Xie *et al*. (2005) uses an inappropriate estimate of π_0 and still has severe under- or over- estimation problem. We have proposed two new modifications to overcome those problems. Simulation studies and application to real data have confirmed that our estimator $\widehat{FDR}(d)_2$ gives significantly better FDR estimates than $\widehat{FDR}(d)_0$ in Xie *et al*. (2005).

Current null statistics are constructed by randomly assigning the "+" or "-" signs to replicates of genes. As a consequence, the number of "+" and "-" signs can be different in this

random assignment. Mean expression levels of the EE genes will always be 0 regardless of the way of assigning the signs. However, when there is an unbalanced number of "+" and "-", the mean expression levels of the DE genes will not be 0, which may cause the null statistics of DE genes to have different distributions from that of the EE genes. Hence, it is intuitive to deduce that if we make the number of "+" and "-" stay balanced, this problem can be avoided. In Pan (2003) and Zhang (2006), they proposed a series of such kind of "balanced" null statistics, which have the same distribution for both the DE and EE genes. It would be interesting to compare the performance of our FDR estimators and estimators based on "balanced" null statistics in the future research.

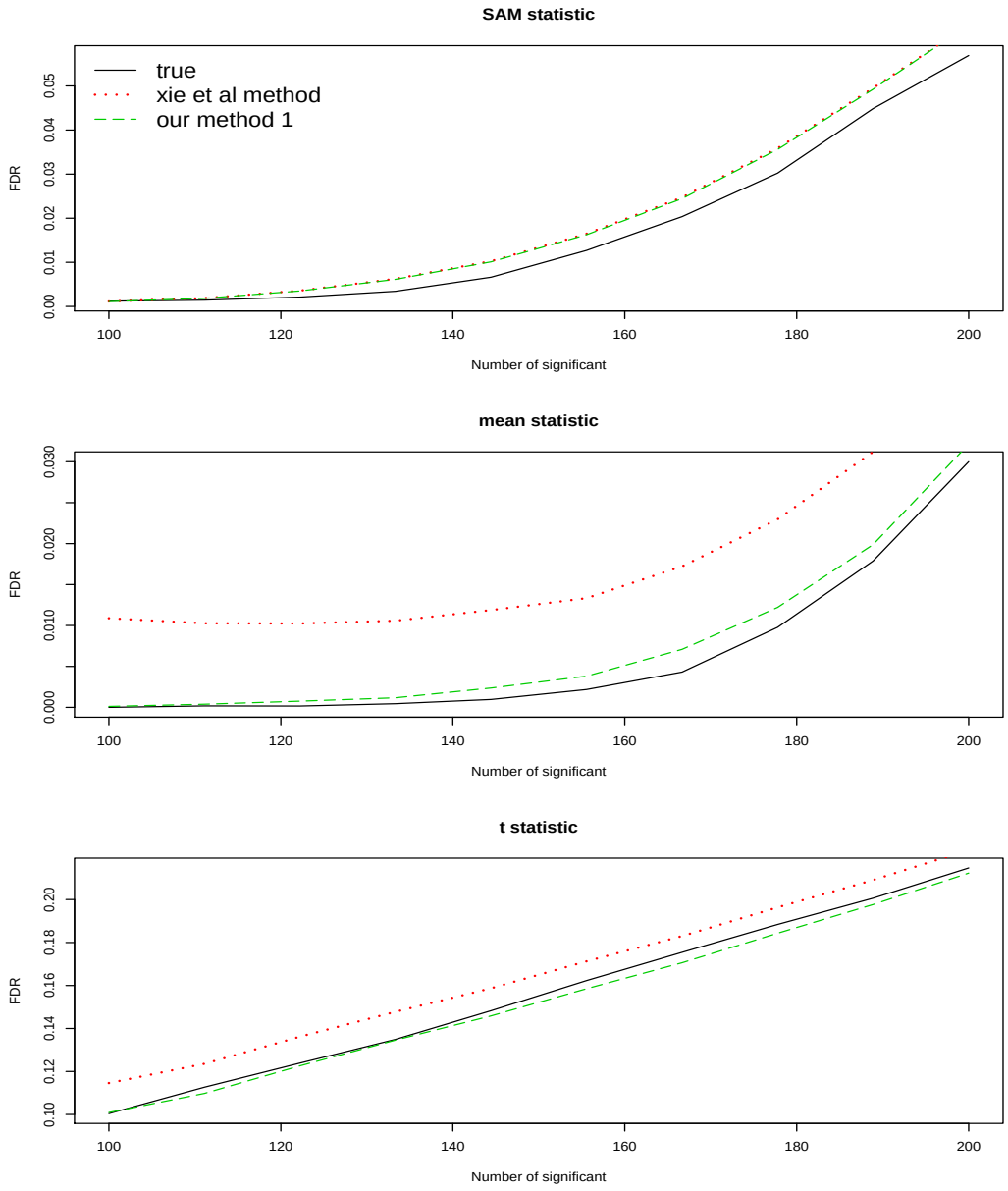

Figure 5.1. The FDR curves of different estimation methods using the SAM, mean, and *t*-statistics. There are 400 DE genes among 4000 genes. The number of claimed significant gene ranges from 100 to 200. $\hat{\pi}_0^{sam}$ is used as the estimate of π_0. Our method 1 is the estimator $\widehat{FDR}(d)_1$ from (5.9).

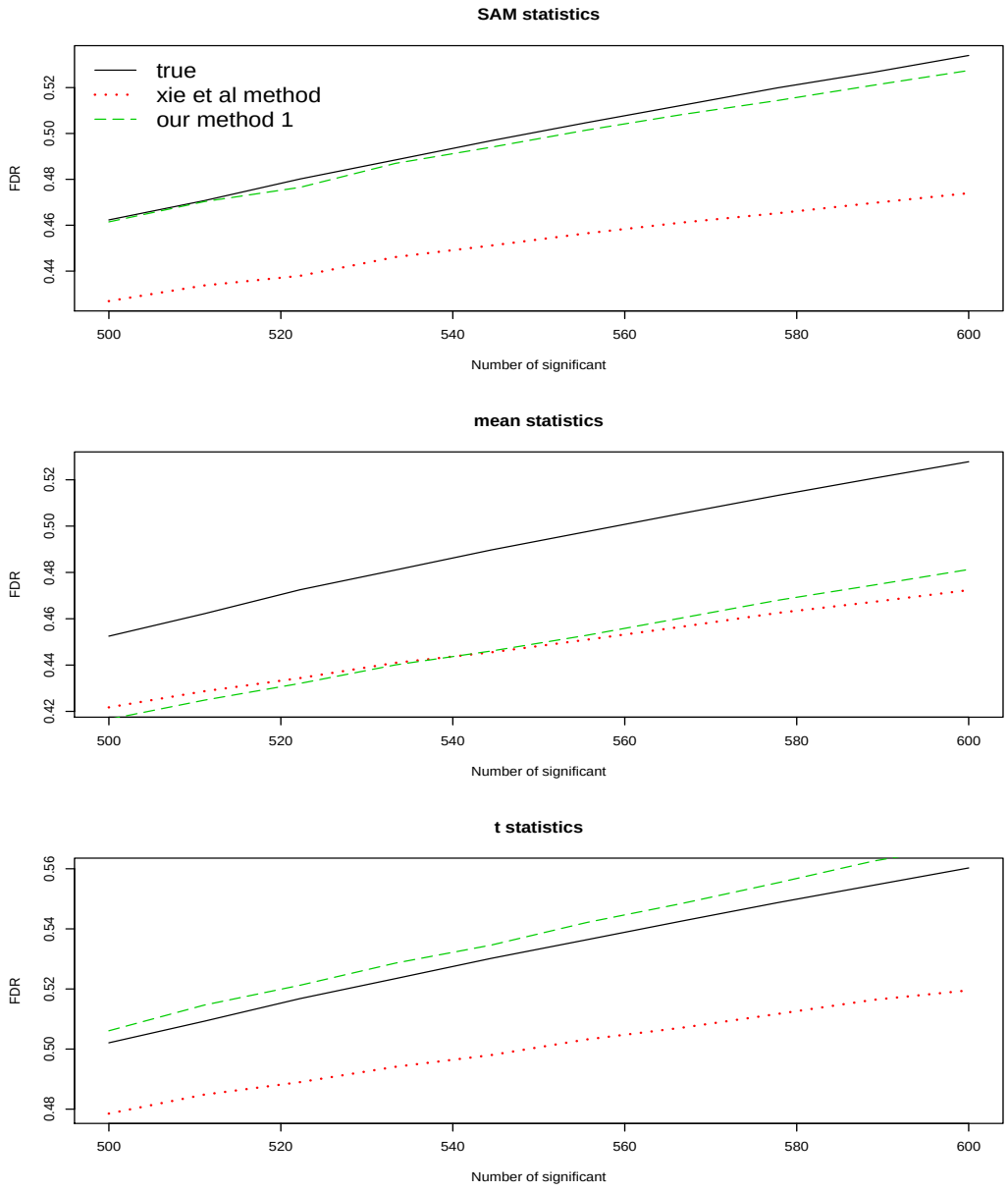

Figure 5.2. The FDR curves of different estimation methods using the SAM, mean, and t-statistics. There are 400 DE genes among 4000 genes. The number of claimed significant gene ranges from 500 to 600. Our method 1 is the estimator $\widehat{FDR}(d)_1$ from (5.9).

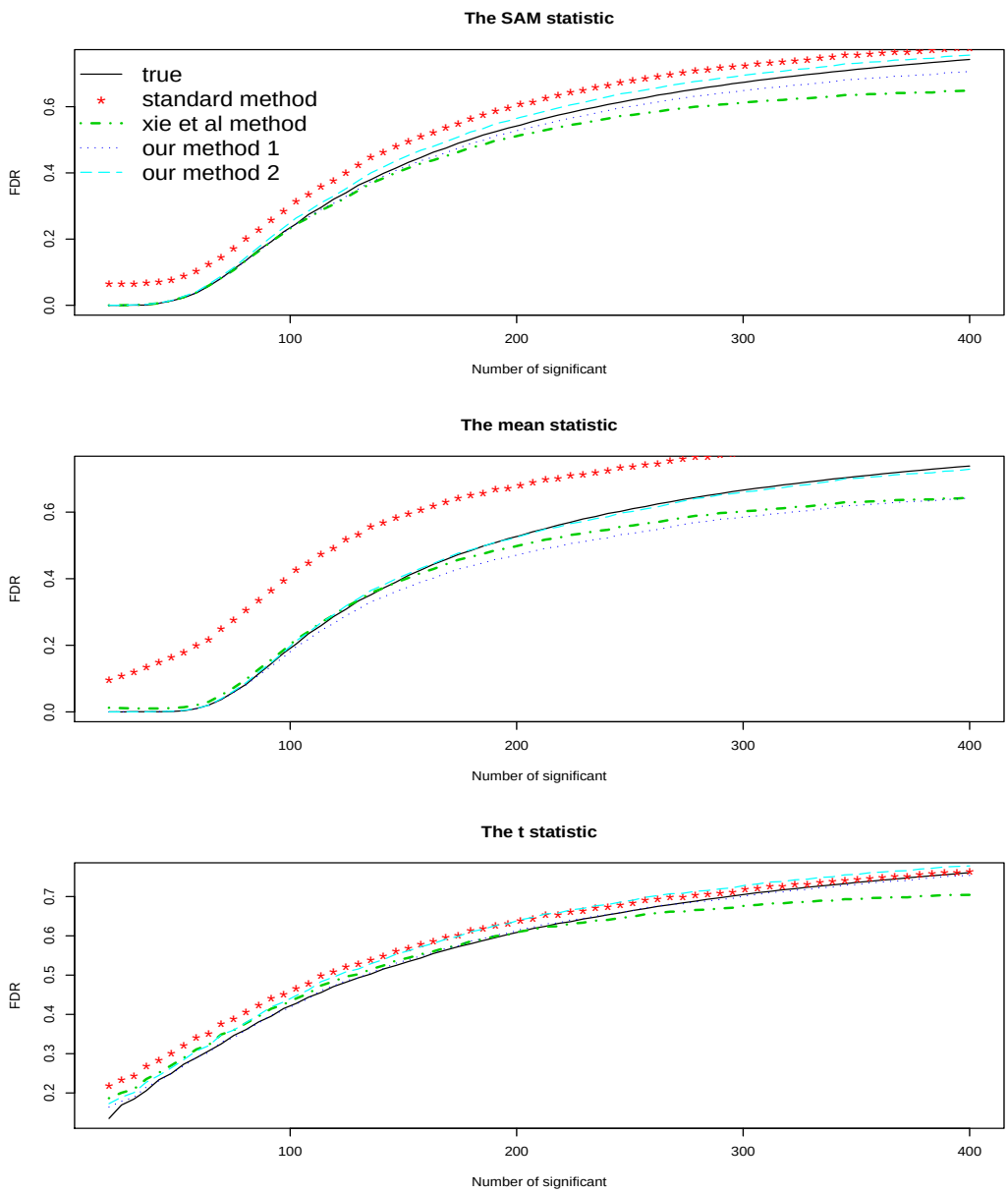

Figure 5.3. The FDR curves of different estimation methods using the SAM, mean, and *t*-statistics. There are 150 DE genes among 4000 genes. The number of claimed significant gene ranges from 20 to 400. $\hat{\pi}_0^{sam}$ is used as estimate of π_0. Our methods 1 and 2 are the estimators $\widehat{FDR}(d)_1$ from (5.9) and $\widehat{FDR}(d)_2$ from (5.10), respectively.

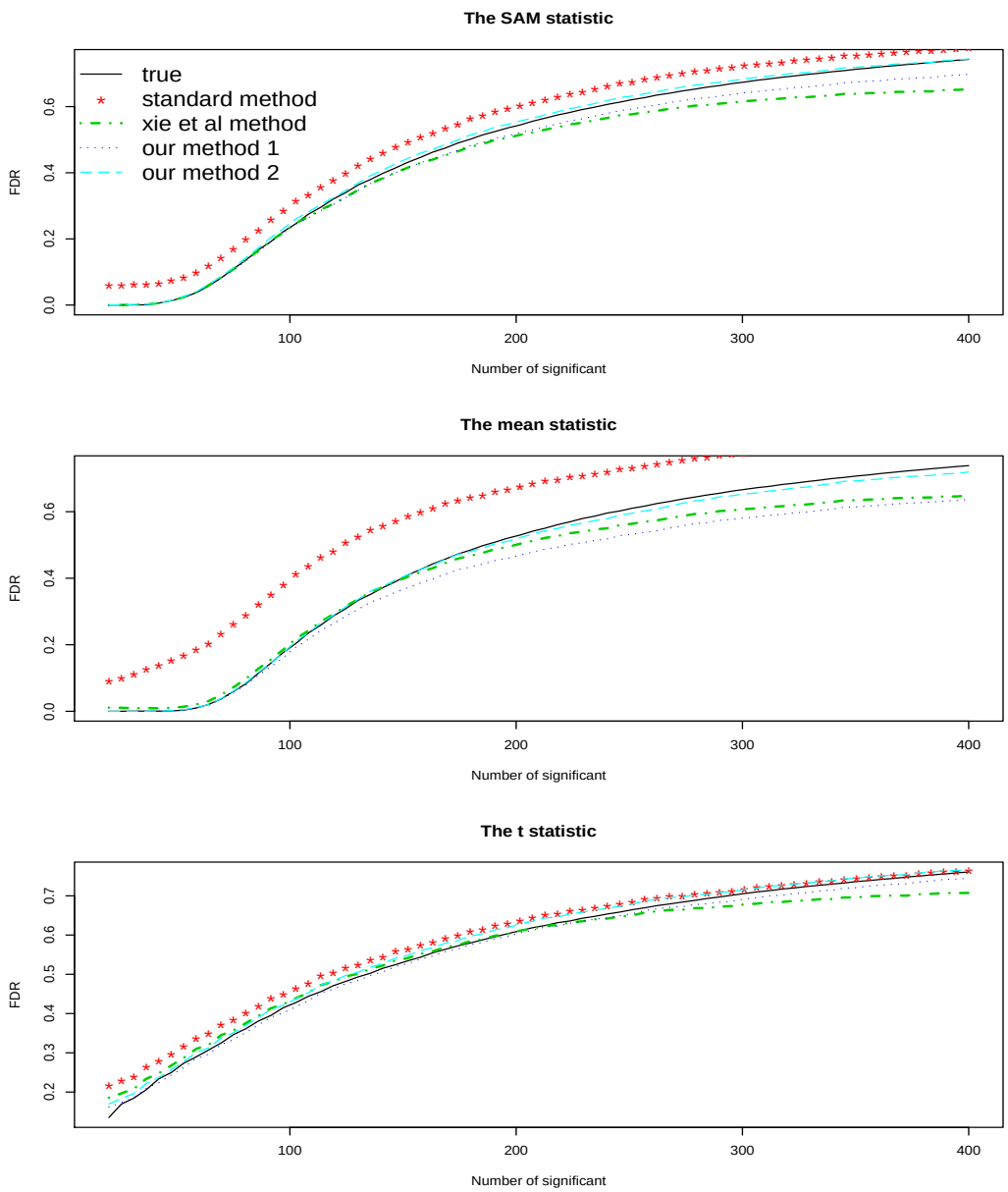

Figure 5.4. The FDR curves of different estimation methods using the SAM, mean, and *t*-statistics. There are 150 DE genes among 4000 genes. The number of claimed significant gene ranges from 20 to 400. The true $\pi_0 = 3850/4000$ is used as estimate of π_0. Our methods 1 and 2 are the estimators $\widehat{FDR}(d)_1$ from (5.9) and $\widehat{FDR}(d)_2$ from (5.10), respectively.

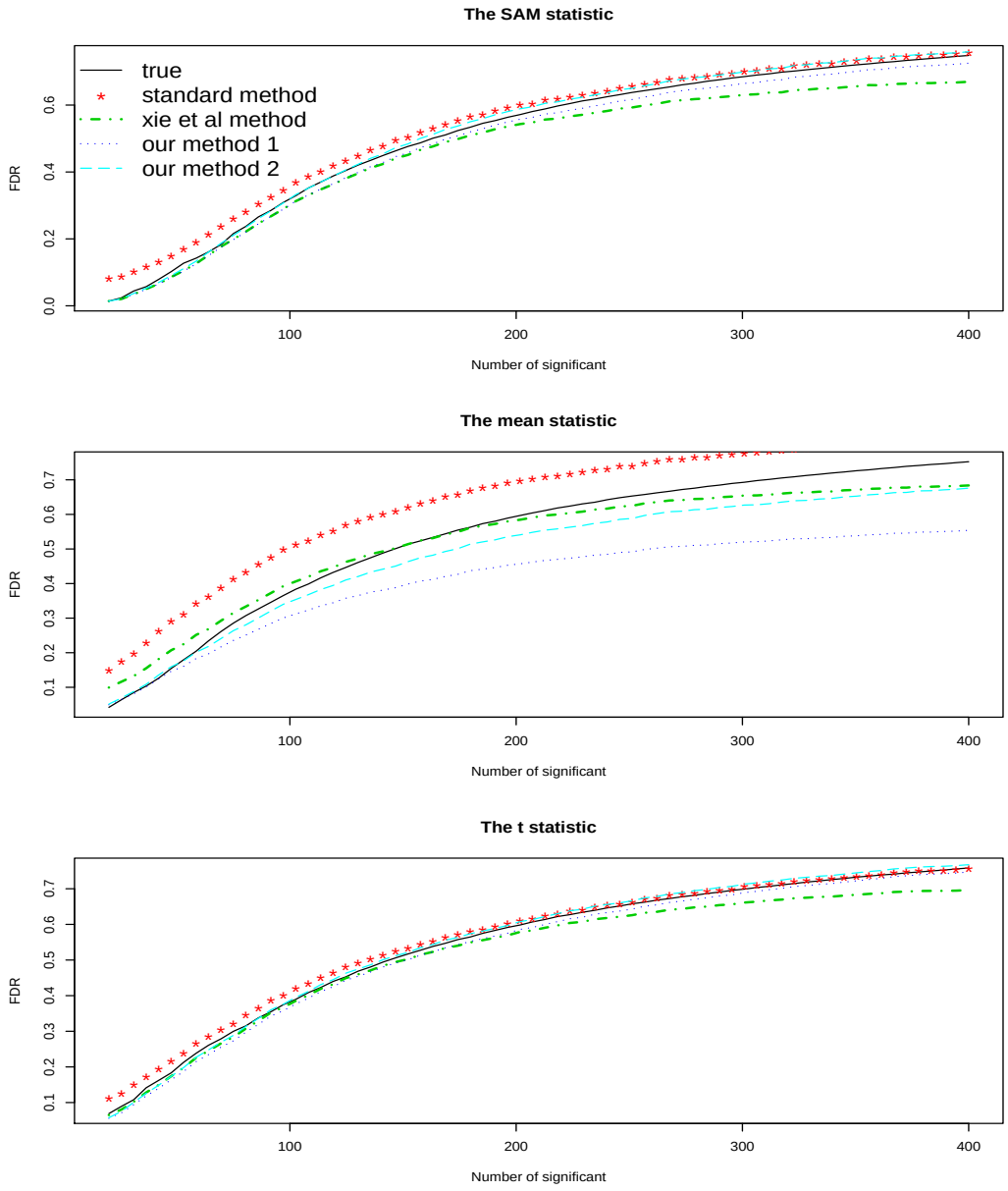

Figure 5.5. The FDR curves of different estimation methods using the SAM, mean, and t-statistics. Mimicking the real data. There are 150 DE genes among 4000 genes. The number of claimed significant gene ranges from 20 to 400. Our methods 1 and 2 are the estimators $\widehat{FDR}(d)_1$ from (5.9) and $\widehat{FDR}(d)_2$ from (5.10), respectively.

CHAPTER 6

ESTIMATING THE PROPORTION OF EQUIVALENTLY EXPRESSED GENES IN MICROARRAY DATA BASED ON TRANSFORMED TEST STATISTICS

In microarray data analysis, the false discovery rate (FDR) is now widely accepted as the control criterion to account for multiple hypothesis testing. The proportion of equivalently expressed genes (π_0) is a key component to be estimated in the estimation of the FDR. Some commonly used π_0 estimators (BUM, SPLOSH, QVALUE, and LBE) are all based on p-values and they are essentially upper bounds of π_0. Simulation shows that these four methods significantly overestimate the true π_0 when differentially expressed genes and equivalently expressed genes are not well separated. To solve this problem, we first introduce a novel way of transforming the test statistics to make them symmetric about 0. Then we propose a π_0 estimator based on the transformed test statistics using the symmetry assumption. Real data and simulation both show that the π_0 estimate from our method is less conservative than BUM, SPLOSH, QVALUE, and LBE in most of the cases. Simulation results also show that our estimator always has the least mean squared error among these five methods.

6.1. Introduction

Microarray technology makes it possible to measure the expression levels of thousands of genes simultaneously. A typical goal of analyzing the gene expression data from this technology is to determine which genes are differentially expressed (DE) between two treatment groups, which is actually a multiple hypothesis testing problem. Controlling family-wise error rate (FWER) is a common practice in regular multiple testing problems. However, due to the large number of genes in microarray data, false discovery rate (FDR) introduced by Benjamini and Hochberg (1995) is now commonly used as the choice of the Type I error rate in microarray studies. It is defined as the expected percentage of false positive genes among the claimed significant genes. It was proved that in many cases controlling FDR is more appropriate compared to controlling FWER since the FDR approaches typically rejects more null hypotheses than the FWER approaches (Yekutieli and Benjamini (1999) and Benjamini and Yekutieli (2001)). Several FDR controlling methods are implemented in the R *multtest* package (Pollard *et al.* (2004)).

To estimate FDR, the proportion of equivalently expressed (EE) genes (π_0) needs to be estimated first. A number of methods have been proposed to estimate π_0 and most of them are based on the distribution of p-values under the null hypothesis. For gene i, the null hypothesis is that gene i is EE, and a p-value (p_i) will be computed. Notice that the p-values of EE genes are uniformly distributed and denote the distribution of p-values of DE genes by $h_1(p)$. It is reasonable to model the overall p-values as a mixture distribution with two components (McLachlan and Peel (2000)):

$$h(p) = \pi_0 * 1 + (1 - \pi_0)h_1(p). \tag{6.1}$$

In Pounds and Morris (2003), the authors proposed a method called BUM using a beta-uniform mixture distribution to approximate $h(p)$. Then they estimated π_0 as $\hat{\pi}_0 = \hat{h}(1)$, which assumed $h_1(1) = 0$ and is an upper bound of the true π_0. Langaas and Lindqvist (2005) adopted the same assumption but used nonparametric maximum likelihood method to estimate $\hat{h}(p)$. SPLOSH (Pounds and Cheng (2004)) uses a local regression technique (LOESS; Cleveland and Devlin (1988)) to fit $h(p)$ and gives $\hat{\pi}_0 = min_p \hat{h}(p)$ as the estimator, which is still an upper bound. Storey and Tibshirani (2003) proposed the QVALUE method. Given a tuning parameter λ, QVALUE estimates π_0 by $\hat{\pi}_0(\lambda) = \frac{\#(p_i > \lambda)}{n(1-\lambda)}$. It can be proved that $\hat{\pi}_0(\lambda) \to \hat{h}(1)$ as $\lambda \to 1$ (Dalmasso *et al.* (2005)), so QVALUE also overestimates π_0 like BUM and SPLOSH. All these estimators work well if the following assumption holds: few p-values of DE genes are close to 1. Otherwise, if this assumption is strongly violated which will happen when DE and EE genes are not well separated, all of them will tend to overestimate. There are other methods not requiring this assumption. Allison *et al.* (2002) proposed a parametric method to estimate π_0. Dalmasso *et al.* (2005) proposed the LBE method based on the moments of p-values, which also only gives an upper bound of π_0. More recently, Lai (2007) proposed a moment based method which requires no distribution assumption. Unfortunately, his method only works well when there are enough replicates (>8). As we can see from above, the commonly used π_0 estimators BUM, SPLOSH, QVALUE and LBE are all actually upper bounds of π_0.

Most of the current π_0 estimators are based on p-values because a p-value is a unified measurement of significance. However, as a result of using p-values, we may lose some nice properties, such as the symmetry and unimodality of the original test statistics from which the p-values are computed. As we know, the commonly used test statistics are t-type statistics, which are generally symmetrically distributed and the use of symmetry can be helpful in estimation of π_0.

Bordes *et al*. (2006) proposed a nonparametric method to estimate the parameters in a two component mixture model with an unknown component, assuming the unknown distribution is symmetric. The authors also tried to apply this method to microarray data by fitting a similar model as (6.1) to test statistics. Since the t-type test statistics (without absolute value) for upregulated and downregulated DE genes obviously have different distributions, they cannot be modeled into one component. Hence, the authors constructed an F-type test statistic and assumed that it has a symmetric density. However, assuming an F-type test statistic to be symmetrically distributed is obviously not a reasonable assumption.

Although the method in Bordes *et al*. (2006) is not very appropriate for microarray data, it inspired us to use test statistics instead of p-values when estimating π_0. In the next section, we will first introduce a transformation to make the test statistics symmetric about 0, then we propose a π_0 estimator based on the transformed test statistics using the symmetry assumption. Some theoretical results are given. Finally, application to real microarray data sets and intensive simulations are conducted to compare the performance of our method with BUM, SPLOSH, QVALUE and LBE.

6.2. Methods

6.2.1. The test statistic and the null statistic

Suppose that Y_{ij} is the expression level of gene *i* in array j ($i = 1, 2, ..., n$; $j = 1, ..., j_1, j_1$+1,..., j_1+j_2), and the first j_1 and last j_2 arrays are obtained under the two different conditions. For gene i, the null hypothesis is that the mean expression levels under the two conditions are the same.

To test this hypothesis, a possible test statistic would be the standard two sample t-statistic. However, it only works well when the normality assumption is not strongly violated, which is not always the case in practice. A class of nonparametric statistical methods (Pan *et al.* (2003), Zhao and Pan (2003), Pan (2003), Zhang (2006)) have been proposed to overcome this problem. The basic idea is to directly estimate the null distribution of the test statistic Z by constructing a null statistic z which has the null distribution of Z. In other words, for EE genes, the test and null statistics have the same distribution. Among those methods, we decided to use the test and null statistics in Zhang (2006) because their performance are robust and they have improved power over other methods. The test and null statistics are as following:

$$Z = \frac{\frac{\overline{Y}_{11}+\overline{Y}_{12}}{2} - \frac{\overline{Y}_{21}+\overline{Y}_{22}}{2}}{s_0 + \sqrt{\frac{\frac{1}{j_{11}}+\frac{1}{j_{12}}}{4} s_1^2 + \frac{\frac{1}{j_{21}}+\frac{1}{j_{22}}}{4} s_2^2}}, \tag{6.2}$$

$$z = \frac{\frac{\overline{Y}_{11}-\overline{Y}_{12}}{2} + \frac{\overline{Y}_{21}-\overline{Y}_{22}}{2}}{s_0 + \sqrt{\frac{\frac{1}{j_{11}}+\frac{1}{j_{12}}}{4} s_1^2 + \frac{\frac{1}{j_{21}}+\frac{1}{j_{22}}}{4} s_2^2}}, \tag{6.3}$$

where $j_{11} = j_{12} = j_1/2$ if j_1 is even, and $j_{11} = j_{12} - 1 = (j_1 - 1)/2$ if j_1 is odd. j_{21} and j_{22} are similarly defined. $\overline{Y}_{11}$,$\overline{Y}_{12}$,$\overline{Y}_{21}$,$\overline{Y}_{22}$ are the sample means of the four partitions of the replicates of each gene under the two experimental conditions. Those four partitions are $(Y_{ij}, j = 1, ..., j_{11})$ and $(Y_{ij}, j = j_{11}+1, ..., j_1)$ from condition 1; $(Y_{ij}, j = j_1+1, ..., j_1+j_{21})$ and $(Y_{ij}, j = j_1 + j_{21} + 1, ..., j_1 + j_2)$ from condition 2. $s_1^2 = \frac{\sum_{j=1}^{j_{11}}(Y_{ij}-\overline{Y}_{11})^2+\sum_{j=j_{11}+1}^{j_1}(Y_{ij}-\overline{Y}_{12})^2}{j_1-2+I(j_1=2)}$, $s_1^2 = \frac{\sum_{j=j_1+1}^{j_1+j_{21}}(Y_{ij}-\overline{Y}_{21})^2+\sum_{j=j_1+j_{21}+1}^{j_1+j_2}(Y_{ij}-\overline{Y}_{22})^2}{j_2-2+I(j_2=2)}$ are the two pooled sample variances from the replicates under each condition. s_0 is a fudge factor invented by SAM (Tusher *et al.* (2001)). In practice, B sets of null statistics are constructed by permutations of data carried within each experimental condition.

6.2.2. Our method

Now for gene i, $i = 1, 2, ..., n$, we have the test statistic Z_i and the null statistic z_i^b from the bth permutation set. Similarly as in (6.1), we try to fit a mixture distribution to the test statistics. As stated in the previous section, the distribution of the test statistics for EE genes is already known - same as the null statistics. However, for DE genes, there are two types - upregulated and downregulated. Unlike the p-values, the t-type test statistics for those two types of DE genes have means of opposite signs. Hence, instead of two subpopulations (EE and DE) of all genes, it is more appropriate to use three subpopulations (EE, upregulated DE, and downregulated DE). With three components in a mixture model, there are a lot more parameters to be estimated. Fortunately, it is actually less problematic than it seems. We will transform the test statistics as following:

The original test statistic for gene i is Z_i. We can create a new set of test statistics X_i, where $X_i = -Z_i$ or Z_i with the same probability 0.5. In other words, we randomly keep or flip the sign of each Z_i. Now take gene i for example: if gene i is EE, then $E(Z_i) = 0 \Rightarrow E(-Z_i) = 0 \Rightarrow E(X_i) = 0$. Therefore, gene i is still EE after transformation; if gene i is DE, we can see that $E(X_i)$=$E(Z_i)$ or $-E(Z_i)$ with same probability 0.5. In other words, for any DE gene i, no matter if it is originally upregulated or downregulated, this DE gene has the same chance of being upregulated or downregulated after the transformation, which indicates that the proportion of upregulated and downregulated DE genes will be the same. It also implies that the means of upregulated and downregulated DE genes after transformation will be the opposite of each other regardless of their original means.

Before the transformation, separate proportion and mean parameters need to be estimated for upregulated and downregulated DE genes. After the transformation, upregulated and downregulated DE genes have the same proportion and opposite means, which reduces the number of parameters by 2. Now we can propose a mixture model for the new test statistic X_i:

$$f(x) = \pi_0 f_0(x) + (1-\pi_0)(g(x+\mu_0)/2 + g(x-\mu_0)/2), \tag{6.4}$$

where $f(x)$ is the density function of test statistics X_i, $f_0(x)$ is the density function of the test statistics for EE genes, $g(x+\mu_0)$ and $g(x-\mu_0)$ are densities for downregulated and upregulated DE genes ($\mu_0 > 0$), respectively. $g(x)$ is assumed to be an even and unimodal density function. Since the test statistic is of t-type, this assumption is reasonable.

Since it is more convenient to estimate the empirical cumulative distribution function (CDF) than density, we can rewrite (6.4) as:

$$F(x) = \pi_0 F_0(x) + (1-\pi_0)(G(x+\mu_0)/2 + G(x-\mu_0)/2), \tag{6.5}$$

where $F(x)$, $F_0(x)$, and $G(x)$ are the corresponding CDF's for $f(x)$, $f_0(x)$, and $g(x)$, respectively.

The next step is to estimate $G(x)$ from (6.5). First, we have

$$G(x+\mu_0) + G(x-\mu_0) = 2(F(x) - \pi_0 F_0(x))/(1-\pi_0),$$

and this implies

(6.6)

$$G(x-2m\mu_0) + G(x-2(m+1)\mu_0) = 2(F(x-(2m+1)\mu_0) - \pi_0 F_0(x-(2m+1)\mu_0))/(1-\pi_0),$$

Denote the LHS of (6.6) by $C(m)$, we have

$$\sum_{m=0}^{m_1}(-1)^m C(m) = C(0) - C(1) + C(2)... + (-1)^n C(n)$$

$$= G(x) + G(x - 2\mu_0) - G(x - 2\mu_0) - G(x - 4\mu_0)$$

$$+ G(x - 4\mu_0)... + (-1)^{m_1} G(x - 2(m_1 + 1)\mu_0)$$

$$= G(x) + (-1)^{m_1} G(x - 2(m_1 + 1)\mu_0) \to G(x) \tag{6.7}$$

as m_1 tends to infinity since $G(x - 2(m_1 + 1)\mu_0) \to 0$ as $m_1 \to \infty$.

Replace $C(m)$ in (6.7) with the RHS of (6.6), we have

$$G(x) = \sum_{m=0}^{\infty}(-1)^m \frac{F(x - (2m + 1)\mu_0) - \pi_0 F_0(x - (2m + 1)\mu_0)}{(1 - \pi_0)/2}. \tag{6.8}$$

Consider the RHS of (6.8) as a function of x, $p(= \pi_0)$, and $\mu(= \mu_0)$, and denote it by

$$M(x; p, \mu) = \sum_{m=0}^{\infty}(-1)^m \frac{F(x - (2m + 1)\mu) - pF_0(x - (2m + 1)\mu)}{(1 - p)/2}$$

so we have $G(x) = M(x; \pi_0, \mu_0)$. We can also define

$$\hat{M}(x; p, \mu) = \sum_{m=0}^{m_1}(-1)^m \frac{\hat{F}(x - (2m + 1)\mu) - p\hat{F}_0(x - (2m + 1)\mu)}{(1 - p)/2} \tag{6.9}$$

as the corresponding estimator for $M(x; p, \mu)$. $\hat{F}(x) = \#(X_i < x)/n$ and $\hat{F}_0(x) = \sum_{b=1}^{B} \#(z_i^b < x)/(Bn)$ are the corresponding empirical CDF's for $F(x)$ and $F_0(x)$; n is the number of genes; B is the number of sets of null statistics; X_i is the test statistic and z_i^b is the bth set of null statistic for gene i. m_1 is a big integer such that $G(x - 2(m_1 + 1)\mu_0)$ and $1 - G(x + 2(m_1 + 1)\mu_0)$ are all very close to 0. In this chapter, it was chosen to be 20.

Recall that $g(x)$ is an even function, so $G(x) + G(-x) = 1$. Following the idea of Bordes *et al*. (2006) and Hunter *et al*. (2007), we define

$$d(x; p, \mu) = (\frac{1-p}{2})^2 (M(x; p, \mu) + M(-x; p, \mu) - 1)^2. \tag{6.10}$$

Notice that in $M(x; p, \mu)$, $(1-p)/2$ is in the denominator, when p is close to 1 it can be problematic. That is the reason we multiplied the factor $(\frac{1-p}{2})^2$ in (6.10), and an estimate for $M(x; p, \mu)$ is

$$\hat{d}(x; p, \mu) = (\frac{1-p}{2})^2 (\hat{M}(x; p, \mu) + \hat{M}(-x; p, \mu) - 1)^2.$$

As we can see, $d(x; p, \mu) = 0$ for any x when $p = \pi_0$ and $\mu = \mu_0$, which indicates that when $p = \pi_0$ and $\mu = \mu_0$,

$$D(p, \mu) = \int_0^\infty d(x; p, \mu) dx = 0. \tag{6.11}$$

Hence, π_0 can be estimated by minimizing

$$\hat{D}(p, \mu) = \int_0^\infty \hat{d}(x; p, \mu) dx \tag{6.12}$$

with respect to p and μ. The integration starts from 0 because $d(x; p, \mu)$ is a symmetric function of x.

Using the above results, the following procedure is proposed to estimate π_0:

(1) Calculate the test statistics Z_i and the null statistics z_i^b using (6.2) and (6.3), $i = 1, ..., n$, $b = 1, ..B$.

(2) Create the new test statistics X_i by randomly keeping or flipping the sign of each Z_i.

(3) Construct an arithmetic sequence with length $J = 50$. Initial term is 0 and last term is 99_{th} percentile of the new test statistic X_i. Denote this arithmetic sequence by x_j,

$j = 1, ..., J$. Then

$$\widehat{MSD}(p, \mu) = \frac{|x_J|}{J} \sum_{j=1}^{J} \hat{d}(x_j; p, \mu) \tag{6.13}$$

is an approximation for $\hat{D}(p, \mu)$ in (6.12) when J is big enough.

(4) Let $p^* = p_{init} \geq \pi_0$. p_{init} can be set to 1 or some known upper bound of π_0.

(5) Let $p^{**} = p^* - \Delta$, where $\Delta > 0$ is a small number. Minimize $\widehat{MSD}(p^*, \mu)$ and $\widehat{MSD}(p^{**}, \mu)$ with respect to μ. $\widehat{MSD}(p, \mu)$ has two local minimums with respect to μ, choose the one at larger μ (we will explain the reason for doing this). Δ is set to 0.01 here.

(6) If $min_\mu \widehat{MSD}(p^{**}, \mu) < min_\mu \widehat{MSD}(p^*, \mu)$, then let $p^* = p^{**}$ and repeat step 5. If not, $\hat{\pi}_0 = p^*$ will be the estimate of π_0.

(7) Repeat step 2-6 for $R = 20$ times and return the average of all $\hat{\pi}_0$'s, which will be the final estimate of π_0.

Unlike the standard optimization procedure, our searching algorithm is conducted on a decreasing and discrete parameter space of π_0 because of two reasons.

(1) Let $g'(x) = g(x + \mu_0)/2 + g(x - \mu_0)/2$ from (6.4) and $\mu_0' = 0$. We have

$$\begin{aligned} f(x) &= \pi_0 f_0(x) + (1 - \pi_0)(g(x + \mu_0)/2 + g(x - \mu_0)/2) \\ &= \pi_0 f_0(x) + (1 - \pi_0)(g'(x + \mu_0')/2 + g'(x - \mu_0')/2). \end{aligned}$$

Hence $f(x)$ is not identifiable - there are two possible μ_0's. Hence, $\widehat{MSD}(p, \mu)$ is small when μ is close to $\mu_0' = 0$ for any p. If we use the standard optimization procedure and search (p, μ) on the whole parameter space, we may get very biased results.

This is also the reason why we choose the local minimum at the larger μ in step 5 of our algorithm - the μ associated with the other local minimum is close to $\mu_0' = 0$.

(2) Our algorithm is more computationally efficient than the standard optimization procedure since it only searches for p's greater than π_0. This can be proved by the following theorem.

Theorem 6.1. *Suppose Θ is the parameter space for (p, μ), and also assume that when $|x|$ is big enough, $f(x)$ and $g(x+\mu_0)/2 + g(x-\mu_0)/2$ in (6.4) are concave upward, then:*

(i) $D(p, \mu) \geq 0$ *is a continuous function on* Θ.

(ii) $min_\mu D(p, \mu) = 0$ *when* $p = \pi_0$.

(iii) *There exist a threshold π_u such that $min_\mu D(p, \mu)$ is a strictly increasing function of p when $p > \pi_u$.*

(iv) $\| \widehat{MSD}(p, \mu) - D(p, \mu) \| \rightarrow 0$ *as* $m_1 \rightarrow \infty$, $J \rightarrow \infty$ *and* $n \rightarrow \infty$

The proof of this theorem is in the Appendix. From Theorem 6.1 (iii) we can see that as long as $p^* > \pi_u$, $min_\mu D(p^*, \mu)$ will be strictly decreasing as p^* decreases until p^* reaches π_u. We can also notice that π_u cannot be less than π_0 because from Theorem 1 (ii), we know that $min_\mu D(p, \mu)$ will reach 0 as p reaches π_0 - it cannot be strictly decreasing anymore. This implies that $\underset{p^* \geq \pi_u}{argmin}(\underset{\mu}{min} D(p, \mu)) = \pi_u \geq \pi_0$. Our algorithm is actually trying to search for $\hat{\pi}_0 = \underset{p^* \geq \pi_u}{argmin}(\underset{\mu}{min} \widehat{MSD}(p, \mu))$, and $\widehat{MSD}(p, \mu)$ converges to $D(p^*, \mu)$ by Theorem 6.1 (iv). Hence, our estimator $\hat{\pi}_0$ will converge to π_u, an upper bound of π_0. Through intensive simulations in the next section, we will show that our $\hat{\pi}_0$ is less conservative than the π_0 estimates given by BUM, SPLOSH, QVALUE and LBE in most of the cases. In fact, π_u is very close to the true π_0 in some scenarios.

6.3. Results

6.3.1. Real Data

First, we will apply our method, along with BUM, SPLOSH, QVALUE and LBE, to two real microarray data sets. The first data set is the leukemia data from Golub *et al*. (1999). In this study, the purpose was to find differentially expressed genes between acute lymphoblastic leukaemia (ALL) and acute myeloid leukaemia (AML) samples. The total number of genes is 7129 and there are 27 replicates for the ALL and 11 replicates for the AML. The data was pre-processed by the method in Pan *et al*. (2002).

The second data set is the breast cancer data from Hedenfalk *et al*. (2001). This paper tried to find genes which were differentially expressed in tumors with $BRCA1$ mutations and tumors with $BRCA2$ mutations. In the original data set, there are 3226 genes, 7 replicates for $BRCA1$ and 8 replicates for $BRCA2$. As suggested by Storey and Tibshirani (2003), 56 genes were removed in our study because they have expression level greater than 20, which were considered not reliable.

Recall that the input for BUM, SPLOSH, QVALUE and LBE are p-values, and the input for our method are test and null statistics. For every data set, test statistic from (6.2) and null statistic from (6.3) were computed. B=100 sets of null statistics were obtained as in Storey and Tibshirani (2003). Then the p-value p_i for every gene i was computed by the method in Storey and Tibshirani (2003) using test and null statistics.

In step 4 of our method, we let $p_{init} = 2\sum_{i=1}^{n} p_i/n$. From (2.13), we have: $h(p) = \pi_0 * 1 + (1-\pi_0)h_1(p) \Rightarrow E(p) \geq 0.5\pi_0 \Rightarrow \pi_0 \leq 2E(p)$. Therefore, the chosen p_{init} satisfies the condition that $p_{init} \geq \pi_0$.

Table 6.1. Comparison of π_0 Estimates from our method, BUM, SPLOSH, QVALUE and LBE for the Golub *et al.* (1999) and Hedenfalk *et al.* (2001) data.

Data	our method	BUM	SPLOSH	QVALUE	LBE
Golub *et al.*	0.560	0.592	0.662	0.652	0.685
Hedenfalk *et al.*	0.533	0.603	0.675	0.709	0.710

Table 6.1 summarizes the π_0 estimate of our method, BUM, SPLOSH, QVALUE and LBE (BUM, SPLOSH, QVALUE and LBE all have R implementations). Among the four methods other than our method, BUM always gives the smallest π_0 estimates while the estimates from the other three methods are close. On the other hand, π_0 estimate from our method is always the smallest. Since the π_0 estimates from all the above methods are essentially upper bounds of π_0, our upper bound is apparently less conservative than the others.

Although the real data application shows some advantage of our method, getting a complete idea about its performance is hard because the true π_0 is unknown. For this reason, intensive simulations with pre-specified π_0 are conducted in the next section.

6.3.2. Simulated Data

Data were generated for n=10,000 genes under two conditions, mimicking the large number of genes in practice. Each condition has four replicates, aiming to study the performance of π_0 estimators when the number of replicates is small (which is usually the case because of the relatively high cost of microarray experiments). π_0=0.4, 0.6 or 0.8, representing small, medium and large proportion of EE genes. There are three types of simulation set-ups corresponding to three different situations:

(a). *EE, DE genes well separated*

For $10000\pi_0$ EE genes, the expression levels were generated from $N(0,1)$ under both conditions. For $10000(1-\pi_0)$ DE genes, the expression levels under condition 1 were also generated from $N(0,1)$; the expression levels under condition 2 were generated from either $N(3,1)$ or $N(-3,1)$, representing upregulated or downregulated DE genes. The ratio $\#$(upregulated genes)/ $\#$(downregulated genes) is 1 for π_0=0.4 and 0.8; when π_0=0.6, the ratio is 2.

(b). *EE, DE genes not well separated*

All the other configurations are exactly the same as set-up (a) except that the expression levels of DE genes under condition 2 are generated from $N(1,1)$ or $N(-1,1)$.

(c). *Mimic the real data*

For EE gene i, the expression levels under both conditions are generated from $N(0,\sigma_i^2)$, where σ_i is generated from $Gamma(2,4)$. For DE gene j, the expression levels are generated from $N(\mu_{1j},\sigma_j^2)$ for condition 1 and from $N(\mu_{2j},\sigma_j^2)$ for condition 2, where μ_{1j} and μ_{2j} are generated from $N(0,2)$, and σ_j is generated from $Gamma(2,4)$.

Next we estimate π_0 using our method, BUM, SPLOSH, QVALUE and LBE for the simulated data in the same way as for the real data. We repeated the simulation and estimation process 100 times for each set-up. The mean, standard error (SE), and mean squared error (MSE) for all the estimators are summarized in Table 6.2 , 6.3 and 6.4, respectively.

For Set-up (a), when DE and EE genes are well separated, there should be few DE genes with p-values around 1, which is the assumption of BUM, SPLOSH, QVALUE and LBE. Hence, they are all expected to give accurate π_0 estimates. In fact, the results confirm that QVALUE, and LBE all give satisfactory results, as well as our method. It is worth noting that these three

Table 6.2. Comparison of the mean and bias of the π_0 estimates from our method, BUM, SPLOSH, QVALUE and LBE for set-up (a), EE, DE genes well separated; (b), EE, DE genes not well separated; and (c), Mimic the real data. The values outside and inside parenthesis are mean and bias,respectively.

Set-up	π_0	our method	BUM	SPLOSH	QVALUE	LBE
(a)	0.8	0.797	0.711	0.899	0.798	0.792
		(-0.003)	(-0.089)	(0.099)	(-0.002)	(-0.008)
	0.6	0.594	0.480	0.828	0.598	0.594
		(-0.006)	(-0.120)	(0.228)	(-0.008)	(-0.006)
	0.4	0.391	0.274	0.725	0.397	0.398
		(-0.009)	(-0.126)	(0.325)	(-0.003)	(-0.002)
(b)	0.8	0.818	0.841	0.842	0.869	0.878
		(0.018)	(0.041)	(0.042)	(0.069)	(0.078)
	0.6	0.636	0.693	0.717	0.739	0.747
		(0.036)	(0.093)	(0.117)	(0.139)	(0.147)
	0.4	0.456	0.568	0.597	0.610	0.623
		(0.056)	(0.168)	(0.197)	(0.210)	(0.223)
(c)	0.8	0.865	0.873	0.872	0.902	0.908
		(0.065)	(0.073)	(0.072)	(0.102)	(0.108)
	0.6	0.728	0.749	0.764	0.792	0.799
		(0.128)	(0.149)	(0.164)	(0.192)	(0.199)
	0.4	0.596	0.620	0.659	0.679	0.695
		(0.196)	(0.220)	(0.259)	(0.279)	(0.295)

Table 6.3. Comparison of the standard error of the π_0 estimates from our method, BUM, SPLOSH, QVALUE and LBE for set-up (a), EE, DE genes well separated; (b), EE, DE genes not well separated; and (c), Mimic the real data.

Set-up	π_0	our method	BUM	SPLOSH	QVALUE	LBE
(a)	0.8	0.004	0.004	0.041	0.020	0.043
	0.6	0.005	0.004	0.039	0.016	0.040
	0.4	0.003	0.004	0.049	0.018	0.035
(b)	0.8	0.032	0.011	0.033	0.023	0.040
	0.6	0.049	0.012	0.029	0.021	0.052
	0.4	0.046	0.006	0.026	0.020	0.040
(c)	0.8	0.021	0.009	0.034	0.022	0.046
	0.6	0.026	0.010	0.032	0.018	0.045
	0.4	0.023	0.010	0.030	0.019	0.044

Table 6.4. Comparison of the mean squared error of the π_0 estimates from our method, BUM, SPLOSH, QVALUE and LBE for set-up (a), EE, DE genes well separated; (b), EE, DE genes not well separated; and (c), Mimic the real data.

Set-up	π_0	our method	BUM	SPLOSH	QVALUE	LBE
(a)	0.8	0.00003	0.00793	0.01052	0.00038	0.00193
	0.6	0.00006	0.01436	0.05377	0.00026	0.00164
	0.4	0.00010	0.01599	0.10795	0.00034	0.00121
(b)	0.8	0.00136	0.00181	0.00284	0.00528	0.00759
	0.6	0.00372	0.00876	0.01459	0.01986	0.02427
	0.4	0.00528	0.02815	0.03931	0.04463	0.05160
(c)	0.8	0.00475	0.00537	0.00631	0.01096	0.01367
	0.6	0.01709	0.02229	0.02774	0.03732	0.04161
	0.4	0.03891	0.04845	0.06783	0.07811	0.08915

methods give an underestimation of π_0 while they are supposed to be conservatively biased. Nevertheless, the underestimation bias are very small so they can be explained by variability. BUM and SPLOSH both give notably biased estimates compared to the other three methods in this set-up. The reason may be that BUM and SPLOSH both need to fit $h(p)$ in (6.1), and the fitted $\hat{h}(p)$ does not approximate the real data well. Except when $\pi_0 = 0.6$ the SE of our method is 0.0001 greater than BUM, the SE and MSE of our method are always the smallest among all the five methods.

DE and EE genes are not well separated in Set-up (b). Therefore, we would expect BUM, SPLOSH, QVALUE and LBE to largely overestimate π_0, which is confirmed by the results. Our method also overestimates π_0, but to a much less degree. Our method also has the smallest MSE for all π_0's. The SE of our method is relatively big when π_0=0.4 and 0.6 but they are still within an acceptable range.

Set-up (c) adds more variations in the simulation process to mimic the real data. As we expected, the bias of all π_0 estimates tend to increase compared to (b) because of the bigger

variation. Nevertheless, our method still gives the least biased estimate and has the smallest MSE compared with the other methods.

6.4. Discussion

In this chapter, we introduce a way of transforming the test statistics, which may be asymmetrically distributed, to make them symmetric about 0. Then we propose a π_0 estimator based on the transformed test statistics using the symmetry assumption. The real data application and simulation results show the advantageous performances of the proposed method compared with BUM, SPLOSH, QVALUE and LBE.

There are several important parameters in our estimation procedure, such as m_1 in (6.9), J in step 3 of our procedure, Δ in step 5, and R in step 7. As we can see, the precision of our estimator will increase as m_1, J, and R increase and as Δ decreases. However, the computational burden will also increase. Hence, more research are necessary to find the optimized choice of those parameters so that we can achieve a balance between precision and computational efficiency.

Furthermore, since this chapter has focused on microarray data with a large number of genes and a small number of replicates, more comprehensive studies are needed to compare the performance of different π_0 estimators under other data configurations.

It should also be noticed that our method is applicable in situations where the test statistics are t-type, hence it is not as general as other methods which are based on p-values.

6.5. Appendix

First, we need a lemma.

Lemma. *Suppose $F(x)$ and $f(x)$ are the CDF and PDF of an even function, respectively; Also assume when $|x|$ is big enough $f(x)$ is concave upward. Define*

$N(x;\mu,F)=\sum_{m=0}^{\infty}(-1)^m(F(x-(2m+1)\mu)+F(-x-(2m+1)\mu))$,

Then there exist a certain threshold $t>0$ such that when $\mu>t$,

$$x>\mu \Leftrightarrow N(x;\mu,F)>1/2$$

$$x<\mu \Leftrightarrow N(x;\mu,F)<1/2.$$

PROOF. We only need to consider $x>0$ since $N(x;\mu)$ is an even function. When $x=\mu$, it is obvious that

$N(x;\mu,F)=\sum_{m=0}^{\infty}(-1)^m(F(-2m\mu)+F(-(2m+2)\mu))$

$=F(0)-F(-2\mu)+F(-2\mu)-F(-4\mu)+F(-4\mu)-...=F(0)=1/2.$

Suppose the corresponding PDF of $F(x)$ is $f(x)$,

$$\frac{\partial N(x;\mu,F)}{\partial x}=\sum_{m=0}^{\infty}(-1)^m(f(x-(2m+1)\mu)-f(-x-(2m+1)\mu)).$$

Consider one part of the RHS of the above equation

$\sum_{m=2k}^{2k+1}(-1)^m(f(x-(2m+1)\mu)-f(-x-(2m+1)\mu))$

$=f(x-(4k+1)\mu)+f(-x-(4k+3)\mu)-f(-x-(4k+1)\mu)-f(x-(4k+3)\mu),$

and from the assumption we know that when $|x|$ is big enough, $f(x)$ is concave. Hence, when μ is big enough,

$f(x-(4k+1)\mu)+f(-x-(4k+3)\mu)>f(-x-(4k+1)\mu)-f(x-(4k+3)\mu)$

for any k.

Therefore $\sum_{m=0}^{\infty}(-1)^m(f(x-(2m+1)\mu)-f(-x-(2m+1)\mu))>0$, which implies

$$\frac{\partial N(x;\mu,F)}{\partial x}>0.$$

Hence, when μ is big enough, $x > \mu \Leftrightarrow N(x;\mu,F) > 1/2$ and $x < \mu \Leftrightarrow N(x;\mu,F) < 1/2$.

$\square$

PROOF. Proof of Theorem 1.

(i) $D(p,\mu) = \int_0^\infty d(x;p,\mu)dx$, and $d(x;p,\mu)$ is continuous and bounded. From Lebesgue dominated convergence Theorem, we can conclude that $D(p,\mu)$ is continuous.

(ii) Since

$$\begin{aligned} d(x;\pi_0,\mu_0) &= (\frac{1-\pi_0}{2})^2(M(x;\pi_0,\mu_0) + M(-x;\pi_0,\mu_0) - 1)^2 \\ &= (\frac{1-\pi_0}{2})^2(G(x) + G(-x) - 1)^2 = 0, \end{aligned}$$

Therefore $D(\pi_0,\mu_0) = \int_0^\infty d(x;\pi_0,\mu_0)dx = 0$.

Also we have $min_\mu D(\pi_0,\mu) \geq 0$ and $min_\mu D(\pi_0,\mu) \leq D(\pi_0,\mu_0) = 0$. Hence, $min_\mu D(\pi_0,\mu) = 0$.

(iii) First, we will prove that if μ is big enough, $D(p,\mu)$ is a strictly increasing function of p.

Since $d(x;p,\mu) = (\frac{1-p}{2})^2(M(x;p,\mu) + M(-x;p,\mu) - 1)^2$, and

$$M(x;p,\mu) = \sum_{m=0}^{\infty}(-1)^m \frac{F(x-(2m+1)\mu) - pF_0(x-(2m+1)\mu)}{(1-p)/2}$$

Denote $K(x) = G(x+\mu_0)/2 + G(x-\mu_0)/2$ and plug the RHS of (6.5) into the above equation, we have

$$\frac{1-p}{2}M(x;p,\mu) = \sum_{m=0}^{\infty}(-1)^m\{(1-\pi_0)K(x-(2m+1)\mu) + (\pi_0 - p)F_0(x-(2m+1)\mu)\}$$

Write $d(x;p,\mu)$ in terms of $N(x;\mu,F)$ in the Lemma, we have

$d(x;p,\mu) = ((1-\pi_0)N(x;\mu,K) + (\pi_0 - p)N(x;\mu,F_0) - (1-p)/2)^2$.

Take the derivative of $d(x;p,\mu)$ w.r.t p, we have:

$$\frac{\partial d(x;p,\mu)}{\partial p} = 2((1-\pi_0)N(x;\mu,K) + (\pi_0 - p)N(x;\mu,F_0) - (1-p)/2)(1/2 - N(x;\mu,F_0)).$$

When μ is big enough and $x > \mu$, we know that $N(x;\mu,K) > 1/2$ and $N(x;\mu,F_0) > 1/2$ from the Lemma. Hence,

$(1-\pi_0)N(x;\mu,K) + (\pi_0 - p)N(x;\mu,F_0) - (1-p)/2 > (1-\pi_0+\pi_0-p-1+p)/2 = 0$,

and $1/2 - BN(x;\mu,F_0) < 0$.

Therefore $\frac{\partial d(x;p,\mu)}{\partial p} < 0$. When $x < \mu$, it can be similarly proved that $\frac{\partial d(x;p,\mu)}{\partial p} < 0$. Hence $\frac{\partial D(p,\mu)}{\partial p} = \int_0^\infty \frac{\partial d(x;p,\mu)}{\partial p}dx < 0$ and $D(p,\mu)$ is a strictly increasing function of p.

We can also see that when $p = 1$, $\underset{\mu}{argmin} D(p,\mu) \to \infty$ and $D(p,\mu)$ is a continuous function. Hence, for any $c > 0$, there exist a p_c such that $\underset{\mu}{argmin} D(p_c,\mu) > c$. Now suppose we have $p^* > p^{**}$, and p^* and p^{**} are big enough such that $\underset{\mu}{argmin} D(p^{**},\mu) > t$, the upper bound in the Lemma, and from the Lemma, we have

$$min_\mu D(p^*,\mu) = D(p^*, \underset{\mu}{argmin} D(p^*,\mu)) < D(p^{**}, \underset{\mu}{argmin} D(p^*,\mu)) \leq min_\mu D(p^{**},\mu).$$

(iii) is proved.

(iv) From Lemma 3.2 (iii) in Bordes and Vandekerkove (2007), (iv) is obviously true.

$\square$

CHAPTER 7

CONCLUSION

7.1. Summary

In this dissertation, a series of methods have been proposed to detect differentially expressed genes and estimate the false discovery rate for the two-condition microarray data. The performance of our methods are evaluated using both simulated and real data.

The t-mixture model in Chapter 3 improves the performance of the normal mixture model method (Pan *et al*. (2003)) by fitting the test and null statistics with a more appropriate, heavy-tailed t-mixture distribution instead of a normal mixture distribution. Because of the heavy-tail property of t-mixture model, we show that only one set of permutation of null statistics is enough for the t-mixture model to achieve satisfactory performance, which saves a lot of computation time compared to the normal mixture model.

Chapter 4 is a natural extension of Chapter 3, in which the t-mixture model is used to build a model-based, continuous FDR estimator. This new model-based FDR estimator does not have the problems of the empirical FDR estimators such as the granularity problem, the zero FDR problem and the nonexistent FDR problem. The model-based FDR has smaller MSE compared to the empirical FDR in the simulation study.

Chapter 5 and 6 are all improvements to the current FDR estimators for microarray data. Chapter 5 improves the current method for estimating the number of false positive genes when all genes are EE (FP^* in (1.3)), a key element in the FDR estimators. In Chapter 5, we show

that including DE genes when estimating FP^* will give overestimates because DE genes have greater variability than EE genes. We also showed that removing all the predicted DE genes when estimating FP^* would give underestimates. Hence, a two-step FDR estimation procedure is proposed which has been shown to have superior performance than the permutation based method.

Chapter 6 improves the current method for estimating the proportion of EE genes (π_0 in (1.3)), which is also an important component when estimating the FDR. Unlike most of the current methods, which use the p-values as the input and assumes zero density of the p-values around 1 for DE genes, we use the transformed test statistics as the input. In this way, we can take advantage of the symmetry property of the test statistics. The comparison results show that our estimator gives more accurate and precise π_0 estimates than several commonly used π_0 estimators.

7.2. Future work

It is noticeable that in this dissertation, both the fitting of the t-mixture model and the minimization procedure to estimate the π_0 are relatively computation intensive, which will be a great obstacle for people to use them. Therefore, it is desirable to improve their efficiency in the future.

In this dissertation, the parameters of the t-mixture model are estimated by maximum likelihood method using the ECM algorithm, which is known to be time-consuming. There are some possible ways to improve the efficiency of the ECM algorithm. For example, because the null statistics are known to have mean 0, if we fix the μ_i's in (3.9) to 0 instead of letting the ECM algorithm to estimate them when fitting the probability density (f_0) for the null statistics, the

convergence speed of the ECM algorithm may be improved. Another interesting topic would be to study the relationship between the number of observations (number of genes) and the convergence speed of the ECM algorithm. For example, if for 10000 genes, the ECM algorithm takes t_{10000} minutes to converge; for 1000 genes, the ECM algorithm takes t_{1000} minutes to converge. If $10 * t_{1000} < t_{10000}$, then we can take a random sample of size 1000 from the 10000 genes and fit the 1000 genes with the ECM algorithm. Repeat this process for 10 times and get the average estimate of all the parameters. In this way, we can save some computation time and probably get similar results as using all 10000 genes.

Instead of fitting the test and null statistics with the t-mixture model, we can use the p-values as the input. In that way, we can fit the density of the p-values with some much faster algorithm compared to the ECM algorithm. One example would be binning method in Ruppert *et al.* (2007), in which they used a B-spline method to fit the density curve of the binned p-values.

In Chapter 5, a two-step procedure is proposed to estimate the FDR. An interesting modification would be to repeat this two-step procedure until the estimated FDR converges. More specifically, we can use the FDR estimate from step 2 to re-adjust the number of genes excluded from the FDR estimation procedure. Then use the rest of the genes to get a new estimate of the FDR. Repeat this process until there is no significant change in the estimated FDR.

In all current simulations, the genes are considered as independent with each other. In the future, we should explore the performance of our methods while the genes are correlated.

Another worth noting fact is that our methods are limited to the two-condition microarray data. Extensions to the multiple conditions and time course data are not trivial because both the t-mixture model and our π_0 estimator are based on a t-type test statistic. This is also a possible topic for the future work.

References

Allison, D.B, Gadbury, G.L., Heo, M., Fernndez, J.R., Lee, C.K., Prolla, T.A., Weindruch, R. (2002) A mixture model approach for the analysis of microarray gene expression data. *Computational Statistics and Data Analysis*, **39**,1-20.

Benjamini, Y. and Hochberg, Y. (1995) Controlling the false discovery rate: a pratical and powerful approach to multiple testing. *J. R. Statist. Soc*, **57**,289-300.

Benjamini, Y. and Yekutieli, D. (2001) The control of the False Discovery Rate in multiple testing under dependency. *Annals of Statistics*, **29(4)**, 1165-1188.

Bordes,L., Delmas,C., and Vandekerkove,P. (2006) Semiparametric estimation of a two-component mixture model when a component is known, *Scand. J. Statist.*, **33**, 733-752.

Bordes,L. and Vandekerkove,P. (2007) Semiparametric two-component mixture model with a known component: A class of asympototically normal estimators, *http://hal.inria.fr/docs/00/17/47/25/PDF/BV-AoS-07.pdf*, working paper.

Broberg, P. (2003). Ranking genes with respect to differential expression. *Genome Biology*, **4**, R41.

Chu G, Narasimhan B, Tibshirani R, Tusher V. (2007) SAM Significance Analysis of Microarrays-Users guide and technical document. http://www-stat.stanford. edu/ tibs/SAM/sam.pdf

Cleveland,W.S. and Dev,S.J (1988) Locally-weighted regression: an approach to regression analysis by local fitting, *J. Am. Stat. Assoc.*, **83**, 596-610.

Dalmasso,C., Broet,P. and Moreau,T. (2005) A simple procedure for estimating the false discovery rate, *Bioinformatics*, **21**, 660-668.

Dempster, A.P., Laird, N.M., Rubin, D.B. (1977) Maximum likelihood estimation from incomplete data via the EM algorithm (with discussion). *J. R. Statist. Soc. B*, **39**, 1-38.

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

Efron, B., Tibshirani, R., Storey, J.D., Tusher, V. (2001) Empirical Bayes analysis of a microarray experiment. *J. AM Stat Assoc.*, **96**, 1151-1160.

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **285**, 531-537.

Guo, X., Pan, W. (2005) Using weighted permutation scores to detect differential gene expression with microarray data. *J. Comput. Biol.* ,**3(4)**, 989-1006.

Hedenfalk, I., Duggan, D., Chen, Y.D., Radmacher, M., Bittner, M., Simon, R., Meltzer, P., Gusterson, B., Esterller, M., Kallioniemi, O.P., Wilfond, B., Borg, A., Trent, J. (2001) Gene-expression profiles in hereditary breast cancer. *New England Journal of Medicine*. **344**,539-544.

Irizarry,R.A., Hobbs,B., Collin,F., Beazer-Barclay,Y.D., Antonellis,K.J., Scherf,U. and Speed,T.P. (2003) Exploration, Normalization, and Summaries of High density Oligonucleotide Array Probe Level Data. *Biostatistics*, **4**, 249-264.

Jiao, S. and Zhang, S. (2008) The t-mixture model approach for detecting differentially expressed genes in microarrays. *Funct. Integr. Genomics*, **8(3)**,181-6.

Jiao, S. and Zhang, S. (2008) On correcting the overestimation of the permutation based false discovery rate estimator. *Bioinformatics*, **24**,1655-1661.

Hunter,D.R., Wang,S., and Hettmansperger,T.P. (2006) Inference for mixtures of symmetric distributions, *Ann. Statist.*, **35**, 224-251.

Kendziorski, C.M., M.A. Newton, H. Lan, and M.N. Gould (2003). On parametric empirical Bayes methods for comparing multiple groups using replicated gene expression profiles. *Statistics in Medicine*, **22**, 3899-3914.

Kerr, M.K., Martin, M. and Churchill, G.A. (2000). Analysis of variance for gene expression microarray data. *J. Comput. Biol.*, **7**, 819-837.

Khodursky, B.A, Peter, J.B, Cozzarelli, R.N., Botstein, D., Brown, O.P. and Yanofsky, C. (2000). DNA Microarray Analysis of Gene Expression in Response to Physiological and Genetic Changes That Affect Tryptophan

Metabolism in Escherichia coli. *Proc. Natl Acad. Sci. USA*, **97**, 22, 12170-12175.

Lai,Y. (2007) A moment-based method for estimating the proportion of true null hypotheses and its application to microarray gene expression data, *Biostatistics*, **8**, 744-755.

Langaas,M. and Lindqvist, B.H. (2005) Estimating the proportion of true null hypotheses, with application to DNA microarray data, *J. R. Statist. Soc.* B, **67**, 555-572.

Liu, C., Rubin, D.B. (1995) ML estimation of the t distribution using EM and its extensions ECM and ECME. *Statistica Sinica*,**5**, 19-39.

Long, A. D., Mangalam, H. J., Chan, B. Y. P., Tolleri, L., Hatfield, W. G., Baldi, P. (2001) Improved statistical inference from stockticker DNA microarray data using analysis of variance and a Bayesian statistical frame work. *The Journal of Biological Chemistry*, **276**, 19937-44.

McLachlan, G.J., Peel, D. (2000) *Finite Mixture Models.* Wiley Series in Probability and Statistics.

McLachlan, G.J, Bean, R.W., Peel, D. (2002) A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics*, **18**, 413-422.

McLachlan, G.J, Bean, R.W., Jones, Ben-Tovim.L. (2005) A simple implementation of a normal mixture approach to differential gene expression in multiclass microarrays. *Bioinformatics*, **22**, 1608-1615.

Newton, M. A., Kendziorski, C. M., Richmond, C. S., Battner, F. R., and Tsui, K. W. (2001). On differentially variability of expression ratios: improving statistical inference about gene expression changes from microarray data. *J. Comput. Biol.*, **8**, 37-52.

Pan, W., Lin, J., Le, C. (2003) Model-based cluster analysis of microarray gene expression data. *Genome Biol.*, **3**, 1-8.

Pan, W., Lin, J., Le, C. (2003) A mixture model approach to detecting differentially expressed genes with microarray data, *Funct Integr Genomics*, **3**, 117-124.

Pan, W. (2003) On the use of permutation in the performance of a class of nonparametric methods to detect differential gene expression, *Bioinformatics*, **19**, 1333-1040.

Pollard, K.S., Dudoit, S. and van der Laan, M.J. (2004) Multiple Testing Procedures: R multtest Package and Applications to Genomics. *U.C. Berkeley Division of Biostatistics Working Paper Series*. Working Paper **164**. http://www.bepress.com/ucbbiostat/paper164.

Pounds,S. and Cheng,C. (2004) Improving the false discovery rate estimation, *Bioinformatics*, **20**, 1737-1745.

Pounds,S. and Morris,S.W. (2003) Estimating the occurrence of false positives and false negatives in microarray studies by approximating and partioning the empirical distribution of p-values, *Bioinformatics*, **19**, 1236-1242.

. R Development Core Team (2008) R: A language and environment for statistical computing. *R Foundation for Statistical Computing*, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Ruppert, D., Nettleton, D., and Hwang, J. T. G. (2007). Exploring the information in p-values for the analysis and planning of multiple-test experiments, *Biometrics*, **63(2)**, 483-95.

Schena M, Shalon D, Davis RW, Brown PO (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, **270**, 467C470.

Smyth, G. K. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, **3(1)**, Article 3.

Smyth, G. K. (2005). Limma: Linear Models for microarray data. In: *Bioinformatics and Computational Biology Solutions using R and Bioconductor*, R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, W. Huber (eds.), Springer, New York, pages 397-420.

Storey,J.D. and Tibshirani,R. (2003) Statistical significance for genome-wide studies, *Proc. Natl. Acad. Sci. USA*, **100**, 9440-9445.

Thomas, J. G., Olson, J. M., Tapscott, S.J., Zhao, L. P. (2001). An efficient and robust statistical modelling approach to discover differentially expressed genes using genomic expression profiles. *Genome Research*, **11**, 1227-1236.

Troyanskaya, O. G., Garber, M. E., Brown, P. O., Botstein, D., Altman, R. B. (2002). Nonparametric methods for identifying differentially expressed genes in mocroarray data. *Bioinformatics*,**18**,1454-61.

Tusher, V.G., Tibshirani, R., Chu, G. (2001) Significant analysis of microarrays applied to the ionizing radiation response. *PNAS*, **98**, 5116-5121.

Xie, Y., Pan, W., and Khodursky, A. (2005) A note on using permutation based false discovery rate estimate to compare different analysis methods for microarray data. *Bioinformatics*, **21**, 4280-4288.

Yekutieli, D. and Benjamini, Y. (1999) Resampling based False Discovery Rate controlling multiple testing procedure for correlated test statistics. *J Statist. Plann. Inference*, **82**, 171-196.

Zhang, S. (2006) An Improved Nonparametric Approach for Detecting Differentially Expressed Genes with Replicated Microarray Data. *Statistical Applications in Genetics and Molecular Biology*, **5**, Iss. 1, Article 30.

Zhang, S. (2007) A comprehensive evaluation of SAM, the SAM R-package and a simple modification to improve its performance. *BMC Bioinformatics*, **8**, 230.

Zhao, Y., Pan, W.(2003) Modified nonparametric approaches to detecting differentially expressed genes in replicated microarray experiments. *Bioinformatics*, **19**, 1046-1054.

Zhong, S., Khodursky, A., Dykhuizen, E.D., and Dean, M.A.(2004) Evolutionary genomics of ecological specialization. *PNAS*, **101**, 11719C11724.

APPENDIX

a. Codes for fitting a t-mixture model

```
#TMM function is the function to detect differentially expressed genes for
   a given significance level alpha
#y is the expression matrix with row as genes and column as replicates
#alpha is the significance level
#Replicates of 1 to j1 columns are under first condition; replicates of j1
   +1 to j1+j2 columns are under second condition
#tol is a tolerance bound for t-mixture model fitting
#Output of this function is a list of rows numbers corresponding to genes

TMM<-function(y,alpha,j1,j2,tol=0.00001)
{n<-dim(y)[1]
data<-as.matrix(y)
j11<-floor(j1/2)
j12<-j1-j11
j21<-floor(j2/2)
j22<-j2-j21

data1<-as.matrix(data[,1:j1])
data2<-as.matrix(data[,(j1+1):(j1+j2)])
data11<-as.matrix(data[,1:j11])
data12<-as.matrix(data[,(j11+1):j1])
data21<-as.matrix(data[,(j1+1):(j1+j21)])
data22<-as.matrix(data[,(j1+j21+1):(j1+j2)])
y11<-apply(data11,1,mean)
y12<-apply(data12,1,mean)
y21<-apply(data21,1,mean)
y22<-apply(data22,1,mean)
s1<-(apply((data11-y11)^2,1,sum)+apply((data12-y12)^2,1,sum))/(j1-2)
s2<-(apply((data21-y21)^2,1,sum)+apply((data22-y22)^2,1,sum))/(j2-2)

#Search for s0
nh<-c()
for(h in 1:99/100)
```

```
{s0<-h/(1-h)
z.zhang<-(y11-y12+y21-y22)*0.5/(sqrt(s1*(0.25/j11+0.25/j12)+s2*(0.25/j21
   +0.25/j22))+s0)
Z.zhang<-(y11+y12-y21-y22)*0.5/(sqrt(s1*(0.25/j11+0.25/j12)+s2*(0.25/j21
   +0.25/j22))+s0)
cutoff<-quantile(abs(z.zhang),1-alpha)
nh<-c(nh,sum(abs(Z.zhang)>cutoff))
}
h0<-median((1:99/100)[nh==max(nh)])
s0<-h0/(1-h0)

z.zhang<-(y11-y12+y21-y22)*0.5/(sqrt(s1*(0.25/j11+0.25/j12)+s2*(0.25/j21
   +0.25/j22))+s0)
Z.zhang<-(y11+y12-y21-y22)*0.5/(sqrt(s1*(0.25/j11+0.25/j12)+s2*(0.25/j21
   +0.25/j22))+s0)

# t mixture data fit function
tmixture.1<-function(y,g,tol)
{errortimes<-0
ind<-1
con<-1
dim<-1
n<-dim(y)[2]
#initial
while(ind&(errortimes<3))
{ind<-0
ybar<-as.vector(apply(y,1, mean))
cov<-(y-ybar)%*%t(y-ybar)/n
loglik<-(-Inf)
sigma<-matrix(rep(cov,g),dim,dim*g)
u<-matrix(NA,dim,g)
p<-rep(1/g,g)
margin<-matrix(NA,g,n)
mu<-matrix(NA,g,n)
cat("Searching␣for␣initial␣values","\n")
for (rep in 1:30)
{
if(g==1){df<-sample(1:50,1)}
if(g>1){df<-c(250,sample(1:100,g-1))}
for (i in 1:g)
{
u[,i]<-ybar
}
# initial margin
```

```
v<-df
for( i in 1:g)
{
sqrtinv<-sqrt(1/as.numeric(sigma[,(dim*i-dim+1):(dim*i)]))
yprime<-as.vector(sqrtinv*(y-u[,i]))
ysquare<-yprime*yprime
margin[i,]<-p[i]*gamma(v[i]/2+dim/2)*(det(as.matrix(sigma[,(dim*i-dim+1):(
    dim*i)])))^(-0.5)/((pi*v[i])^(dim/2)*gamma(v[i]/2)*(1+ysquare/v[i])^(v[i
    ]/2+dim/2))
mu[i,]<-(v[i]+dim)/(v[i]+ysquare)
}
sum.margin<-apply(margin,2,sum)
if(loglik<sum(log(sum.margin)))
{u0<-u
margin0<-margin
v0<-df
loglik<-sum(log(sum.margin))}
}
u<-u0
margin<-margin0
v<-v0
stop<-c(loglik-1,loglik)
count<-1
cat("Using␣ECM␣algorithm␣to␣estimate␣parameters........","␣\n")


while ( ((abs(stop[count+1]-stop[count])>tol)&con&(count<20))|(((stop[count
    +1]-stop[count])>tol)&con&(count>=20)))
{ptm<-proc.time()
#e step 1

tau<-matrix(NA,g,n)
taumu<-matrix(NA,g,n)
for( i in 1:g)
{tau[i,]<-margin[i,]/sum.margin
taumu[i,]<-tau[i,]*mu[i,]
}
u0<-u
sigma0<-sigma
v0<-v
p0<-p
margin0<-margin
stop0<-stop
```

```
#m step 1

t1<-apply(taumu,1,sum)
p<-apply(tau,1,mean)
for(i in 1:g)
{u[,i]<-as.vector(taumu[i,]%*%t(y))/t1[i]
yu<-t(y-u[,i])*sqrt(taumu[i,])
sigma[,(dim*i-dim+1):(dim*i)]<-t(yu)%*%yu/as.numeric(t1[i])}

# e step 2

margin<-matrix(NA,g,n)
mu<-matrix(NA,g,n)
for( i in 1:g)
{
sqrtinv<-sqrt(1/as.numeric(sigma[,(dim*i-dim+1):(dim*i)]))
yprime<-as.vector(sqrtinv*(y-u[,i]))
ysquare<-yprime*yprime
margin[i,]<-p[i]*gamma(v[i]/2+dim/2)*(det(as.matrix(sigma[,(dim*i-dim+1):(
    dim*i)])))^(-0.5)/((pi*v[i])^(dim/2)*gamma(v[i]/2)*(1+ysquare/v[i])^(v[i
    ]/2+dim/2))
mu[i,]<-(v[i]+dim)/(v[i]+ysquare)
}

sum.margin<-apply(margin,2,sum)
tau<-t(t(margin)/sum.margin)
sum.tau<-apply(tau,1,sum)

# m step 2
for (i in 1:g)
{
constant<-(1/sum.tau[i])*sum(tau[i,]*(log(mu[i,])-mu[i,]))+1+digamma(v[i]/
    2+dim/2)-log(v[i]/2+dim/2)
if (constant<(-0.002926826))
{
solve.v<-function(v){-digamma(v/2)+log(v/2)+constant}
v[i]<-uniroot(solve.v,c(2.680138e-304,2.146436e+14))$root
}else{cat("loop",count, "v",i,">342","\n")
con<-0}
}


margin<-matrix(NA,g,n)
mu<-matrix(NA,g,n)
```

```
for( i in 1:g)
{
sqrtinv<-sqrt(1/as.numeric(sigma[,(dim*i-dim+1):(dim*i)]))
yprime<-as.vector(sqrtinv*(y-u[,i]))
ysquare<-yprime*yprime
margin[i,]<-p[i]*gamma(v[i]/2+dim/2)*(det(as.matrix(sigma[,(dim*i-dim+1):(
    dim*i)])))^(-0.5)/((pi*v[i])^(dim/2)*gamma(v[i]/2)*(1+ysquare/v[i])^(v[i
    ]/2+dim/2))
mu[i,]<-(v[i]+dim)/(v[i]+ysquare)
}
sum.margin<-apply(margin,2,sum)
loglik<-sum(log(sum.margin))
stop<-c(stop,loglik)
count<-count+1

if(is.na(loglik))
{ind<-1
errortimes<-errortimes+1
cat("Not␣converge.␣Try␣another␣initial␣value␣\n")
break}
}
}
if(!ind){return(u0,v0,sigma0,p0,stop0,margin0,loglik)}else{return(loglik=(-
    Inf),margin0=matrix(NA,g,n))}
}


y.z.zhang<-t(as.matrix(z.zhang))

bic.find<-function(fit,k)
{margin<-fit$margin
loglik<-fit$loglik
value<--2*loglik+(k*dim(margin)[1]-1)*log(dim(margin)[2])
return(value)
}

bic<--1
bic0<-0
i<-1
fit.z.zhang.t<-0
while ((i==2)|(bic<bic0))
{

bic0<-bic
```

```
fit.z.zhang.t0<-fit.z.zhang.t

cat("Try␣cluster␣number␣=",i,"\n")

fit.z.zhang.t<-try(tmixture.1(y.z.zhang,i,tol),TRUE)
if(is.matrix(fit.z.zhang.t$margin0)){bic<-bic.find(fit.z.zhang.t,4)}
i<-i+1
}
fit.t.u<-rep(NA,10)
fit.t.v<-rep(NA,10)
fit.t.sigma<-rep(NA,10)
fit.t.p<-rep(NA,10)
fit.t.u[1:length(fit.z.zhang.t0$u0)]<-fit.z.zhang.t0$u0
fit.t.v[1:length(fit.z.zhang.t0$v0)]<-fit.z.zhang.t0$v0
fit.t.sigma[1:length(fit.z.zhang.t0$sigma0)]<-fit.z.zhang.t0$sigma0
fit.t.p[1:length(fit.z.zhang.t0$p0)]<-fit.z.zhang.t0$p0
fit<-fit.z.zhang.t0

fit.t<-function(x){-alpha+sum(fit$p*pt((x-fit$u)/sqrt(fit$sigma),df=fit$v))
   +1-sum(fit$p*pt((-x-fit$u)/sqrt(fit$sigma),df=fit$v))}

critic.t<-uniroot(fit.t,c(-100,100))$root

sig.gene<-(1:n)[abs(Z.zhang)>abs(critic.t)]
return(sig.gene)
}
```

b. Codes for comparison between the model based FDR and the empirical FDR

```
rep<-0
kk<-c()
trynumber<-1000
FDR.mix.t<-fdr.sam<-c(NA,trynumber)
i=1

for (critic in quantile(abs(Z.sam), 1-trynumber:1/7129))
{
rep<-rep+1
p.t<-fit.t.p[i,][!is.na(fit.t.p[i,])]
u.t<-fit.t.u[i,][!is.na(fit.t.u[i,])]
sigma.t<-fit.t.sigma[i,][!is.na(fit.t.sigma[i,])]
v.t<-fit.t.v[i,][!is.na(fit.t.v[i,])]
fit.t.new<-function(x){sum(p.t*pt((x-u.t)/sqrt(sigma.t),df=v.t))+1-sum(p.t*
   pt((-x-u.t)/sqrt(sigma.t),df=v.t))}
```

```
p.t.Z<-fit.t.p[i+1,][!is.na(fit.t.p[i+1,])]
u.t.Z<-fit.t.u[i+1,][!is.na(fit.t.u[i+1,])]
sigma.t.Z<-fit.t.sigma[i+1,][!is.na(fit.t.sigma[i+1,])]
v.t.Z<-fit.t.v[i+1,][!is.na(fit.t.v[i+1,])]
fit.t.new.Z<-function(x){sum(p.t.Z*pt((x-u.t.Z)/sqrt(sigma.t.Z),df=v.t.Z))
   +1-sum(p.t.Z*pt((-x-u.t.Z)/sqrt(sigma.t.Z),df=v.t.Z))}

fdr.sam[rep]<-pi0sam[(i+1)/2]*median(apply(matrix(abs(z.sam)>abs(critic)
   ,10,7129),1,sum))/sum(abs(Z.sam)>abs(critic))
FDR.mix.t[rep]<-pi0sam[(i+1)/2]*fit.t.new(-critic)/fit.t.new.Z(-critic)
}

plot(1:trynumber, fdr.sam,axes=FALSE,type='l', lty=3,col=2,ylim=c(0,0.06),
   ylab="FDR",xlab="Number␣of␣Significant␣Genes")
axis(1,at=c(0,200,400,600,800,1000),labels=c(1000,800,600,400,200,0))
axis(2)
lines(1:trynumber,FDR.mix.t,lty=5,col=3)
legend('topright',c("empirical␣method","our␣method"),lty=c(3,5),col=c(2,3),
   bty='n')
```

c. Codes for comparison between the two-step FDR estimator and the standard method

```
s.stat<-function(y)
{
n<-dim(y)[1]
k<-dim(y)[2]
y.mean<-apply(y,1,mean)
y.var<-apply(y,1,var)
y.var0<-median(y.var)
s.stat<-y.mean*sqrt(k)/(sqrt(y.var)+s0.sam)
return(s.stat)
}

mean.stat<-function(y)
{
n<-dim(y)[1]
k<-dim(y)[2]
y.mean<-apply(y,1,mean)
return(y.mean)
}

t.stat<-function(y)
{
n<-dim(y)[1]
k<-dim(y)[2]
```

```
y.mean<-apply(y,1,mean)
y.var<-apply(y,1,var)
s.stat<-y.mean*sqrt(k)/sqrt(y.var)
return(s.stat)
}




postscript(file="fdr1fdr2_09625.eps", onefile=FALSE, horizontal=FALSE)
par(mfrow=c(3,1))


lengtho<-70
fdr.true=fdr.std=fdr.xie=fdr.zhang=fdr.jiao=fdr.jiao2=matrix(NA,50,lengtho)
for( rep in 1:50)
{
y<-yall[,(rep*5-4):(rep*5)]
n<-dim(y)[1]
k<-dim(y)[2]
nonnull<-150

x.group<-rep(1,k)
data.sam=list(x=y,y=x.group, geneid=as.character(1:nrow(y)),genenames=paste
    ("g",as.character(1:nrow(y)),sep=""), logged2=TRUE)
samr.obj<-samr(data.sam,resp.type="One␣class")

pi0.sam<-min(1,samr.obj$pi0)
s0.sam<-samr.obj$s0
cat(pi0.sam)
nperm<-samr.obj$nperms.act

y0<-y
stat.test<-s.stat(y0)

B<-nperm
perm.m<-matrix(sample(c(1,-1),size=k*B,replace=TRUE,prob=c(0.5,0.5)),k,B)
ysquare<-apply(y0^2,1,sum)
y.per.sum<-y0%*%perm.m
y.var.m<-(ysquare-y.per.sum^2/k)/(k-1)
y.var0<-apply(y.var.m,2,median)
stat.null<-(y.per.sum/sqrt(k))/(sqrt(y.var.m)+s0.sam)
from<-20
to<-400
```

```
sig.number<-seq(from,to, length.out=lengtho)
fnnum=tpnum=fpnum.true=fpnum.est=fpnum.std=fpnum.est.temp=fpnum.est1=fpnum.
   est2=rep(NA,lengtho)

for(i in 1:lengtho)
{
sig.num<-sig.number[i]
quant<-quantile(abs(stat.test),probs=c(1-sig.num/n))
tpnum[i]<-sum(abs(stat.test)>quant)
fpnum.true[i]<-sum((1:n)[abs(stat.test)>quant]<(n-nonnull+1))
fpnum.std[i]<-sum(abs(stat.null)>quant)/B
fpnum.est[i]<-sum(abs(stat.null)[abs(stat.test)<=quant]>quant)/B
fdr.temp<-(fpnum.est[i]*n*pi0.sam)/((n-tpnum[i])*tpnum[i])
quant.temp<-quantile(abs(stat.test),probs=c(1-sig.num*(1-fdr.temp)/n))
fpnum.est.temp[i]<-sum(abs(stat.null)[abs(stat.test)<=quant.temp,]>quant)/B
}
fdr.true[rep,]<-fpnum.true/tpnum
fdr.xie[rep,]<-fpnum.est/tpnum
fdr.std[rep,]<-fpnum.std*(n-nonnull)/(n*tpnum)
fdr.zhang[rep,]<-(fpnum.est*n*0.9625)/((n-tpnum)*tpnum)
fdr.jiao[rep,]<-(fpnum.est.temp*n*0.9625)/((n-tpnum*(1-fdr.zhang[rep,]))*
   tpnum)
}

fdr.true.ave<-apply(fdr.true,2,mean)
fdr.xie.ave<-apply(fdr.xie,2,mean)
fdr.std.ave<-apply(fdr.std,2,mean)
fdr.zhang.ave<-apply(fdr.zhang,2,mean)
fdr.jiao.ave<-apply(fdr.jiao,2,mean)

plot(sig.number, fdr.true.ave,type='l', lty=1,xlab='Number␣of␣significant',
    ylab='FDR',main='SAM␣statistics',col=1)
lines(sig.number, fdr.std.ave, pch='*',cex=1.5, type='p',col=2)
lines(sig.number, fdr.xie.ave, lty=4,lwd=2,col=3)
lines(sig.number, fdr.zhang.ave, lty=3,col=4)
lines(sig.number, fdr.jiao.ave, lty=5,col=5)
legend('topleft',cex=1.6,c("true","standard␣method","xie␣et␣al␣method","our
   ␣method␣1", "our␣method␣2"),col=c(1,2,3,4,5), lty=c(1,NA,4,3,5),lwd=c
   (1,1,2,1,1),pch=c(NA,'*',NA,NA,NA),bty='n')

lengtho<-70
fdr.true=fdr.std=fdr.xie=fdr.zhang=fdr.jiao=fdr.jiao2=matrix(NA,50,lengtho)
for( rep in 1:50)
{
```

```
y<-yall[,(rep*5-4):(rep*5)]
n<-dim(y)[1]
k<-dim(y)[2]
nonnull<-150
x.group<-rep(1,k)
data.sam=list(x=y,y=x.group, geneid=as.character(1:nrow(y)),genenames=paste
    ("g",as.character(1:nrow(y)),sep=""), logged2=TRUE)
samr.obj<-samr(data.sam,resp.type="One␣class")
pi0.sam<-min(1,samr.obj$pi0)
s0.sam<-samr.obj$s0
cat(pi0.sam)
nperm<-samr.obj$nperms.act
y0<-y
stat.test<-mean.stat(y0)
stat.test.sam<-s.stat(y0)
B<-nperm
perm.m<-matrix(sample(c(1,-1),size=k*B,replace=TRUE,prob=c(0.5,0.5)),k,B)
ysquare<-apply(y0^2,1,sum)
y.per.sum<-y0%*%perm.m
y.var.m<-(ysquare-y.per.sum^2/k)/(k-1)
y.var0<-apply(y.var.m,2,median)
stat.null<-y.per.sum/k
from<-20
to<-400
sig.number<-seq(from,to, length.out=lengtho)
fnnum=tpnum=fpnum.true=fpnum.est=fpnum.est.xie=fpnum.std=fpnum.est.temp=
    fpnum.est1=fpnum.est2=rep(NA,lengtho)

for(i in 1:lengtho)
{
sig.num<-sig.number[i]
quant<-quantile(abs(stat.test),probs=c(1-sig.num/n))
quant.sam<-quantile(abs(stat.test.sam),probs=c(1-sig.num/n))
tpnum[i]<-sum(abs(stat.test)>quant)
fpnum.true[i]<-sum((1:n)[abs(stat.test)>quant]<(n-nonnull+1))
fpnum.std[i]<-sum(abs(stat.null)>quant)/B
fpnum.est[i]<-sum(abs(stat.null)[abs(stat.test)<=quant]>quant)/B
fpnum.est.xie[i]<-sum(abs(stat.null)[abs(stat.test.sam)<=quant.sam]>quant)/
    B
fdr.temp<-(fpnum.est[i]*n*pi0.sam)/((n-tpnum[i])*tpnum[i])
quant.temp<-quantile(abs(stat.test),probs=c(1-sig.num*(1-fdr.temp)/n))
fpnum.est.temp[i]<-sum(abs(stat.null)[abs(stat.test)<=quant.temp,]>quant)/B
}
fdr.true[rep,]<-fpnum.true/tpnum
```

```
fdr.xie[rep,]<-fpnum.est.xie/tpnum
fdr.std[rep,]<-fpnum.std*(n-nonnull)/(n*tpnum)
fdr.zhang[rep,]<-(fpnum.est*n*0.9625)/((n-tpnum)*tpnum)
fdr.jiao[rep,]<-(fpnum.est.temp*n*0.9625)/((n-tpnum*(1-fdr.zhang[rep,]))*
   tpnum)
}

fdr.true.ave<-apply(fdr.true,2,mean)
fdr.xie.ave<-apply(fdr.xie,2,mean)
fdr.std.ave<-apply(fdr.std,2,mean)
fdr.zhang.ave<-apply(fdr.zhang,2,mean)
fdr.jiao.ave<-apply(fdr.jiao,2,mean)

plot(sig.number, fdr.true.ave,type='l', lty=1,xlab='Number␣of␣significant',
    ylab='FDR',main='mean␣statistics',col=1)
lines(sig.number, fdr.std.ave, pch='*',cex=1.5, type='p',col=2)
lines(sig.number, fdr.xie.ave, lty=4,lwd=2,col=3)
lines(sig.number, fdr.zhang.ave, lty=3,col=4)
lines(sig.number, fdr.jiao.ave, lty=5,col=5)


lengtho<-70
fdr.true=fdr.std=fdr.xie=fdr.zhang=fdr.jiao=fdr.jiao2=matrix(NA,50,lengtho)
for( rep in 1:50)
{
y<-yall[,(rep*5-4):(rep*5)]
n<-dim(y)[1]
k<-dim(y)[2]
nonnull<-150

x.group<-rep(1,k)
data.sam=list(x=y,y=x.group, geneid=as.character(1:nrow(y)),genenames=paste
   ("g",as.character(1:nrow(y)),sep=""), logged2=TRUE)
samr.obj<-samr(data.sam,resp.type="One␣class")
pi0.sam<-min(1,samr.obj$pi0)
s0.sam<-samr.obj$s0
cat(pi0.sam)
nperm<-samr.obj$nperms.act
y0<-y
stat.test<-t.stat(y0)
stat.test.sam<-s.stat(y0)
B<-nperm
perm.m<-matrix(sample(c(1,-1),size=k*B,replace=TRUE,prob=c(0.5,0.5)),k,B)
ysquare<-apply(y0^2,1,sum)
```

```
y.per.sum<-y0%*%perm.m
y.var.m<-(ysquare-y.per.sum^2/k)/(k-1)
y.var0<-apply(y.var.m,2,median)
stat.null<-(y.per.sum/sqrt(k))/sqrt(y.var.m)
from<-20
to<-400
sig.number<-seq(from,to, length.out=lengtho)
fnnum=tpnum=fpnum.true=fpnum.est=fpnum.est.xie=fpnum.std=fpnum.est.temp=
   fpnum.est1=fpnum.est2=rep(NA,lengtho)

for(i in 1:lengtho)
{
sig.num<-sig.number[i]
quant<-quantile(abs(stat.test),probs=c(1-sig.num/n))
quant.sam<-quantile(abs(stat.test.sam),probs=c(1-sig.num/n))
tpnum[i]<-sum(abs(stat.test)>quant)
fpnum.true[i]<-sum((1:n)[abs(stat.test)>quant]<(n-nonnull+1))
fpnum.std[i]<-sum(abs(stat.null)>quant)/B
fpnum.est[i]<-sum(abs(stat.null)[abs(stat.test)<=quant]>quant)/B
fpnum.est.xie[i]<-sum(abs(stat.null)[abs(stat.test.sam)<=quant.sam]>quant)/
   B
fdr.temp<-(fpnum.est[i]*n*pi0.sam)/((n-tpnum[i])*tpnum[i])
quant.temp<-quantile(abs(stat.test),probs=c(1-sig.num*(1-fdr.temp)/n))
fpnum.est.temp[i]<-sum(abs(stat.null)[abs(stat.test)<=quant.temp,]>quant)/B
}
fdr.true[rep,]<-fpnum.true/tpnum
fdr.xie[rep,]<-fpnum.est.xie/tpnum
fdr.std[rep,]<-fpnum.std*(n-nonnull)/(n*tpnum)
fdr.zhang[rep,]<-(fpnum.est*n*0.9625)/((n-tpnum)*tpnum)
fdr.jiao[rep,]<-(fpnum.est.temp*n*0.9625)/((n-tpnum*(1-fdr.zhang[rep,]))*
   tpnum)
}

fdr.true.ave<-apply(fdr.true,2,mean)
fdr.xie.ave<-apply(fdr.xie,2,mean)
fdr.std.ave<-apply(fdr.std,2,mean)
fdr.zhang.ave<-apply(fdr.zhang,2,mean)
fdr.jiao.ave<-apply(fdr.jiao,2,mean)

plot(sig.number, fdr.true.ave,type='l', lty=1,xlab='Number␣of␣significant',
    ylab='FDR',main='t␣statistics',col=1)
lines(sig.number, fdr.std.ave, pch='*',cex=1.5, type='p',col=2)
lines(sig.number, fdr.xie.ave, lty=4,lwd=2,col=3)
lines(sig.number, fdr.zhang.ave, lty=3,col=4)
```

```
lines(sig.number, fdr.jiao.ave, lty=5,col=5)

dev.off()
```

d. Codes for estimating π_0 using our method

```
pi0.out.leuk<-rep(NA,50)
for (num in 1:50)
{
cat("######################Replic",num,"###################","\n")
Z.zhang<-Z.zhang.leuk
z.zhang<-z.zhang.leuk
m<-7129
B<-100
Z.zhang.new<-Z.zhang*sample(c(-1,1),m,replace=TRUE)
att_1<- abs(Z.zhang)
att0_1 <- abs(z.zhang)
v_1 <- c(rep(T,m),rep(F,m*B))
v_1 <- v_1[rev(order(c(att_1,att0_1)))]
u_1 <- 1:length(v_1)
w_1 <- 1:m
p_1 <- (u_1[v_1==TRUE]-w_1)/(B*m)
pvalue2_1 <- p_1[rank(-att_1)]

F.cdf<-ecdf(Z.zhang.new)
f0.cdf<-ecdf(z.zhang)
F.cdf.n<-ecdf(-Z.zhang.new)
f0.cdf.n<-ecdf(-z.zhang)
iter.num<-20

pi0<-0.6
mse.pi0<-c()
mse.com<-100000
mse.com1<-99999
while ((pi0>0)&(mse.com1<mse.com))
{
mse.com<-mse.com1
opt<-optimize(find.mu,c(0,quantile(Z.zhang.new,0.999)))
mse.pi0<-c(mse.pi0,opt$objective)
cat(c(pi0,opt$objective,opt$minimum),"\n")
pi0<-pi0-0.01
mse.com1<-opt$objective
}
pi0.out.leuk[num]<-pi0+0.02
}
```